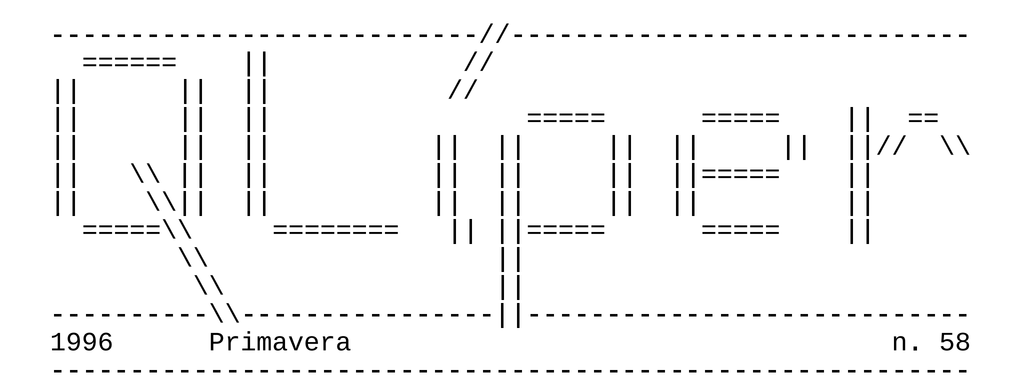


1996 Primavera n. 58

Estamos en el año 16 Después de Sinclair. Toda Hispania está ocupada por las legiones de PCs. ¿Toda? ¡No! Un puñado de irreductibles QLs resiste todavía y siempre ante el primitivo invasor...

QLíper
Redactor: Salvador Merino
Tel. (95) 2475043 (Trabajo)
(95) 2474887 (Casa)
Cerámicas Mary
Ctra. Cádiz (Torreblanca del Sol)
29640 FUENGIROLA (MALAGA)
ESPAÑA

INFORMACION SOBRE QLIPER

Compilación de colaboraciones y distribución: Salvador Merino
Portada: Pedro Reina

El fanzine QLIPER es un medio de comunicación e intercambio de ideas y conocimientos entre usuarios de ordenadores Sinclair QL y compatibles QDOS. El fanzine se realiza mediante la colaboración desinteresada de todas las personas que lo desean.

Indice

Editorial

Este número he preferido dedicarselo al legendario ordenador Sinclair SPECTRUM grabando un emulador Spectrum para QL, otro para PC, y algunos de los mejores programas/juegos comercializados hace más de 12 años.

Si alguien desea comprar una QXL 68040 con 8 Mbytes, Dasio Carballeira vende la suya.

Actualmente estoy escribiendo un programa de gestión inmobiliaria utilizando mi lenguaje MERINO dBASE TIL, que funciona en QL y PC. Consiste en una super base datos que guarda toda la información de una inmobiliaria. Naturalmente, no es para mi, es para un amigo que desea un programa sencillo y muy barato que le solucione su problema.

Por primera vez en toda la historia de nuestro Club, tengo miedo de no poder compilar una nueva revista QLIPER con un minimo de material elaborado por los miembros de nuestro Club. Mi objetivo más inmediato es llegar al número 60 de nuestra revista en disco QLíper. Luego con vistas al año 1997, mucho me temo que la salida de nuestra revista QLíper sea semestral, o de salida irregular. La principal causa: El número de colaboradores/lectores será inferior a DIEZ, y yo ya no me creo capaz de hacer nuevos milagros. Sin embargo, la supervivencia del QL está garantizada gracias a emuladores de su hardware en ordenadores AMIGA, PC, ATARI ST/TT y APPLE MAC (e incluso ya se habla de gente que usan emuladores de QL bajo emuladores de ATARI ST en PCs bajo sistema operativo LINUX), y sistemas operativos compatibles QDOS como el SMSQ.

Salvador Merino, 5 de marzo de 1996

Respuestas

Sobre INFOVIA

Estimado Pedro Reina, lo único que sé de INFOVIA es que es un sistema de comunicación via MODEM de Telefónica a un precio muy barato. Muchas revistas que vendo en mi tienda han regalado discos y CD-ROM conteniendo el software necesario para un PC. No he adquirido aún el software por solamente un motivo: aunque tengo un PC y un MODEM lo suficientemente rápidos para poder usar la red INFOVIA, es necesario un minimo de 8 Mbytes de RAM y ampliar mi modesto y saturado disco duro de 170 Mbytes por uno más grande (y al paso que van los acontecimientos, habria que pensar en comprar un nuevo PC basado en PENTIUM con bus PCI, en vez de ampliar el viejo).
Sobre Internet, hay que decir que es muy atractivo su nuevo look multimedia comparado con su modo teletipo anterior todavía vigente, pero conociendo la mala calidad de la linea de telefónica no es de extrañar nada que el envio de esas hermosas pantallas con sonido sea cuestión de muchisimos minutos de espera a 14400 baudios, pues hay que tener en cuenta que estamos hablando de envios/recepciones de Mbytes. Si he de ser sincero, conociendo mi propia experiencia de años con mi antiguo MODEM de 2400 y ahora mi FAX/MODEM de 14400, no me he atrevido a instalar el software Internet en mi PC por miedo a una escandalosa factura de telefónica, pues ya me dió una vez por visitar varias veces la superlenta red IBERTEXT donde no vi nada de mi interés, y no vea como subió la factura del teléfono.
En resumen, últimamente lo único que hago con mi FAX/MODEM es enviar cartas o documentos digitalizados o escaneados via FAX. El MODEM casi no lo uso, y es muy posible que termine comprando una máquina FAX.

Salvador Merino, 22/3/1996

Ofertas

Dasio Carballeira Tella tf.981 565974/544017
Bo.As Salvadas 29 piso 3 "D"
15705 Santiago de Compostela

26 de febrero de 1996

Queridos amigos:
Podría iniciar esta carta con alguna referencia como "puesto el pie en el estribo" de Cervantes o "Lasciate ogni speranza" del Dante Alighieri, pero prefiero "Freude schöner Göterfunken, Töchter aus Elysium" de la oda a la alegría de Schiller.
Prefiero evocar la alegría de ver que existe un grupo entusiasta, con capacidad para discernir lo fundamentel de lo accesorio, liderado por la inagotable ilusión de Salvador Merino.
Pues aunque el QL tiene ciertas limitaciones especialmente en presentación de textos, fué un adelantado a su tiempo, con medios muy modestos, y que sigue sirviendo perfectamente paraa quién quiere eficacia sin florituras con medios muy modestos. Por eso pienso que todavía tiene futuro para quienes buscan especialmente eficacia en particular en el uso de bases de datos y hoja electrónica, que tienen poco que envidiar para trabajos usuales a los carísimos programas como office professional, excepto, lo mismo que en tratamiento de textos en materia de presentación.
El caso es que mi hija y mi nieto viven inmersos en sistemas basados en el uso del PC, por lo que mi tarjeta QXL de 8 megas les resulta un mundo extraño.
por otra parte mi estado de salud agravado aceleradamente me hace casi totalmente imposible manejar el ordenador
Por eso he decidido poner en venta algunas de mis piezas de hardware.
Entre otros:
tarjeta para PC QXl con 8 megas de ram
la expansión de memoria Expandaram 512
un Joystick
Ql Toolkit II en Rom
dispositivo para pronunciar textos en inglés:Qtalk de Maxtronics.
incluyendo: Alien Hijack
Puzzlemania
Pointer's & writers toolkit de Jochen Merz programa original
Qsys system utilities for the QL de Jochen Merz programa original

Si alguien pudiera estar interesado puede contactar conmigo o incluso si lo considera, realizar sus ofertas por teléfono o por carta a la dirección y teléfonos indicados.

Os deseo salud e ilusión para seguir adelante.

Noticias

EL SISTEMA OLIMPO HA SIDO EL CUARTO CLASIFICADO EN UN CONCURSO DE PC ACTUAL

El sistema OLIMPO v2.0 de Pedro Reina ha conseguido la cuarta clasificación en un concurso de programación en C en la revista PC ACTUAL. El premio ha sido un paquete Borland C++ 4.5 and Database Tools.

Se habian recibido 50 programas para concursar, pero solamente 17 fueron seleccionados como finalistas. La clasificación final fue:

1 Premio.- Declarado desierto.
2 Premio.- TIM (Tratamiento de imágenes en movimiento) diseñado por Daniel Rodríguez Martínez y José Miguel Contreras.
3 Premio.- RGRAFW (Aplicación de representación de funciones bajo WINDOWS) diseñado por Juan M. Ramos Heredero.
4/5 Premio:
- Synasia (simulador de robots) diseñado por Roberto Sofín Ballano.
- OLIMPO diseñado por Pedro Reina.

Según mi opinión, OLIMPO deberia haber ganado el primer premio. Si no lo ha obtenido, como dice Pedro Reina, ha sido por carencia de colores y ratón (compatibilidad total con el QL). Sin embargo, Pedro está muy satisfecho con la clasificación obtenida.

Salvador Merino, 19 de enero de 1996

Otros Temas

LA LLEGADA DEL AÑO 2000 OBLIGARA DATAR DE NUEVO EL PARQUE MUNDIAL DE ORDENADORES CON UN COSTE DE 78 BILLONES.
==

Esta noticia apareció en el periódico ABC con fecha 11-1-96.

Nunca un fallo informático tan insignificante ha pasado una factura económica tan alta. Setenta y ocho billones de pesetas costará, según las primeras estimaciones, conseguir que el parque mundial de ordenadores reconozca la llegada del año 2000. Los actuales programas informáticos han sido diseñados desde hace décadas para que reconozcan las fechas con los últimos dos dígitos de tal forma que para los ordenadores el año 2000 será el año 00. IBM creará un equipo de diez mil técnicos que se dedicará exclusivamente a solventar este inesperado problema.

El problema es antiguo. Con la llegada de la fecha fatídica, los relojes internos de los PCs, se ajustarán automáticamente a enero de 1980, cuando fueron diseñados e introducidos los sistemas de datación de estos ordenadores (nadie esperaba que un sistema como el IBM PC estuviese activo en el año 2000). Actualmente ya existen en el mercado diversos programas informáticos destinados a solucionar este problema.

A los usuarios de QXL no sé en que medida nos puede afectar el problema de datación de los PCs. Por lo pronto en mi QXL con la última versión del sistema operativo SMSQ es posible cambiar la fecha al año 2050.

En un SINCLAIR QL el reloj/calendario comienza en el año 1961 (Tio Clive Sinclair fundó su primera empresa, Sinclair Radionics, en 1962) y termina en el año 2029. Yo la primera vez que lei esa limitación solamente se me ocurrió pensar "¡Qué poco tiempo!", y luego "bueno, aún faltan 43 años para que ocurra eso, y no hay que preocuparse pues nunca actualizo la fecha y hora cuando pongo en marcha mi QL" (era 1986).
En el caso del QL el problema de la hora/calendario del 2029 se podria solucionar con un simple parcheo del sistema operativo. En un PC la cosa se complica, porque hay que actualizar el hardware y software a la vez.

En resumen, quedan menos de 4 años para solucionar un desastre de dimensiones abismales en la mayoria de las empresas que usen PCs compatibles (especialmente en Bancos, Seguros, Seguridad Social, Bancos de Sangre, etc ....). Sin embargo, parece que el verdadero problema es actualizar el viejo software PC que trabaja con una fecha de solamente dos dígitos.

Salvador Merino, 19 de enero de 1996

Programas contenidos en este disco

*** Spectator 1.85 *** (Versión utilizable con QXL)
Spectator_Xample_boot
Spectator_exe

Spectator_txt
Updates_txt
SpecTech_txt
B2SPT_bas
Speculator_bas
Z80_Zombies
Z80_Pool
Z80_Raiders
Z80_Pinball
Z80_Cauldron
Z80_Arkanoid
Z80_CastDoom
Z80_Commando
Z80_Bounder
Z80_DynDan2
Z80_Throttle
Z80_Smashout
Z80_TLL
Z80_Bomber
Z80_Kong2
*** Z80 3.03 ***
Z80-303.ZIP (probablemente el mejor emulador Spectrum 48/128K para PC)

Documentos en su formato original:

--- Apuntes de F. Alonso ---
apuntes1_doc
apuntes2_doc
glosario_doc
disposit_doc
--- Comentarios P. Reina ---
Comentarios_txt

Comentarios a febrero de 1996

1. Todo es personal
2. Olimpo a concurso
3. Borland C++ 4.5 and Database Tools
4. Programación para Microsoft Windows
5. Curso de la U.N.E.D.
6. Mi uso del PC
7. Mi futuro
8. El proyecto para Oxford University Press
9. Infovía

Todo es personal

Normalmente cuando escribo para esta maravillosa revista suelo contar cosas bastante objetivas, referentes sobre todo a programación. Sin embargo, hoy quiero escribir de un modo completamente subjetivo, porque me apetece explayarme un poco con vosotros, mis amigos.

Por tanto, todo lo que viene a continuación son exclusivamente opiniones personales, que no pretenden polemizar sobre ningún punto, sino sólo expresar en voz alta mis valoraciones personales.

Olimpo a concurso

Como sabéis, llevo ya tres años desarrollando el sistema de programación Olimpo, del que han aparecido aquí muchos comentarios sobre él y programas escritos con él. También sabéis que está escrito exclusivamente en C.

Pues bien, la revista PC Actual organizó el año pasado un concurso de programación en C, del que yo me enteré gracias a que Salvador me mandó una carta proponiéndome que mandara Olimpo. Desde el primer momento Salvador pensó que Olimpo tendría alguna posibilidad en el concurso. Yo no estaba tan seguro, porque normalmente en los concursos se busca espectacularidad y Olimpo de eso no tiene absolutamente nada. De todas formas, me pareció una buena excusa para intentar terminar la versión 2.0.

Me puse a trabajar y al fin lo pude presentar, con un manual muy cuidado que iba maquetado con PageMaker, dentro del plazo establecido, en septiembre. Ahora tocaba esperar. El resultado del concurso se retrasó un par de meses y, aunque no esperaba nada, tenía esa cosilla en el estómago.

Me fui de vacaciones de Navidad a Galicia y al volver había dos mensajes en el contestador: uno de Salvador felicitándome por el premio y otro de PC Actual para que les llamara para recoger lo que me había correspondido. Al día siguiente me caí de la cama y fui directamente al quiosco. Estupendo: de los tres premios principales del concurso, el primero había quedado desierto, y de los dos accesits, a mí me habían dado uno. De un total de más de 50 programas presentados al concurso, sólo dos habían superado a Olimpo. Lo tomé como un reconocimento al mucho trabajo que llevaba realizando con mucha seriedad, así que me animó mucho pensar que aunque Olimpo sólo maneja cuatro colores y ni siquiera maneja ratón, un grupo de informáticos había visto algo bueno en él.

El premio que me dieron es el paquete de desarrollo "Borland C++ 4.5 and Database Tools".

Borland C++ 4.5 and Database Tools

Viene en un enorme cajón de 40 cm de largo, ocupado completamente por libros, ya que el software viene en un CD-ROM. La documentación ocupa 12 libros y unos cuantos cuadernillos. Además, hay más documentación en el CD-ROM.

Si se instalara todo en el disco duro, ocuparía unos 120 Mb. Sin embargo, yo lo tengo instalado para que se ejecute directamente desde el CD-ROM, de modo que no ha ocupado más allá de 10 ó 15 megas.

El compilador de C++ admite todas las posibilidades nuevas del C++, que están todavía en desarrollo. Yo no las uso, puesto que sólo programo en C. Se pueden generar ejecutables para MS-DOS, Windows de 16 bits y Windows de 32 bits. Yo sólo lo estoy usando para crear programas para Windows de 16 bits.

Se incluyen dos herramientas útiles para desarrollar programas: un depurador (debugger) y un analizador de rendimiento (profiler). Aunque ya había visto un depurador, puesto que la versión de Turbo C++ que uso habitualmente lleva uno incorporado, para mí era nuevo el analizador. Éste me parece una herramienta muy interesante, mucho más que el depurador.

El depurador se usa sólo cuando se programa mal, es decir, cuando hay errores. Pero el analizador se usa cuando se quiere mejorar el rendimiento del programa; se encarga de informar del tiempo que tarda cada línea del programa, cuántas veces se llama en el programa, etc. Todavía no la he usado, pero creo que me va a resultar de interés.

Cuando se programa para Windows en C o en C++ hay fundamentalmente tres posibilidades:

* Programar en C haciendo llamadas directas a los servicios que proporciona Windows. Son unos 1500 y constituyen lo que se denomina el API de Windows (Application Program Interface).

* Programar en C++ utilizando las clases que proporciona Microsoft en su librería de clases (MFC, Microsoft Fundation Classes).

* Programar en C++ con la librería de clases creada por Borland (OWL, Object Windows Library)

Dado que Borland y Microsoft son enemigos a muerte, en este paquete no se incluye la MFC, sino sólo la OWL, y, por supuesto, documentación del API.

Para manejar bases de datos el paquete incluye una librería (la Borland Database Engine) que admite manejar bases de datos en formatos dBase y Paradox.

El paquete tiene algunas cosillas más, de menor importancia.

En conjunto, es excelente. No en vano en el mercado tiene un precio de 70000 pesetas más IVA.

Programación para Microsoft Windows

Llevaba bastante tiempo pensando que debía empezar a aprender a programar para Windows, incluso me había comprado un libro para intertarlo, pero me parecía tan rollo que nunca lo había conseguido.

Claro, tampoco tenía compilador y así era difícil. Pero con BC++ en casa, ya tenía más atractivo.

Ya he aprendido los rudimentos, y lo que es mejor, le he perdido el miedo. Sólo programo usando el API de Windows, en C, directamente, pero aún así tengo acceso a todas las posibilidades de Windows. Aunque son muchas funciones las que hay, están documentadas y fácilmente accesibles. Es cuestión de leer y probar un poco. Además, se va ampliando el horizonte poco a poco, no hace falta hacerlo todo de golpe.

Me da la impresión, ahora que conozco un poco mejor Windows por dentro, de que es muy plasta de programar y muy ineficiente cuando se ejecuta. Como hago siempre, intentaré usar lo bueno y olvidar lo malo.

Curso de la U.N.E.D.

En mi trabajo como profesor de instituto tengo que justificar al menos 100 horas de cursos cada seis años para poder tener derecho a un dinero extra. Esto se llama cobrar sexenios, y son independientes de los trienios. ¡Vaya morro tenemos los profes! Bueno, pues esas 100 horas se pueden conseguir asistiendo a unos cursillos penosos que organiza el propio ministerio, pero esos cursillos son de 20 ó 30 horas, de modo que hay que hacer varios y desplazarse muchas tardes, cosa que a mí no me apetecía.

Así que me apunté a un curso organizado por la U.N.E.D. que cuesta 35000 pesetas pero cuenta como 120 horas y no hay que ir a ningún sitio. Elegí un curso que me pudiera servir de algo: uno de iniciación a la programación bajo Windows.

Me hicieron comprar un libro de 10000 pesetas, que encima no es de la última versión de Windows sino de la anterior. Empezamos bien. Bueno, el curso se extiende teóricamente de diciembre de 1995 a junio de 1996, pero yo ya lo he terminado a mediados de febrero. Un nivel ínfimo.

Pedían para aprobar el curso dos cosas: responder por escrito unas preguntas y escribir unos programas. Las preguntas son de vergüenza, porque son a nivel de usuario de Windows y elementales. Los programas son once, y se dan unas indicaciones para ir realizándolos. Son penosos, no sirven para nada y ¡algunas indicaciones son erróneas! La repera. He tenido que tomármelo con calma, para no pensar que he tirado 45000 pesetas. Realmente he hecho una inversión: he comprado mi primer sexenio. Me temo que en España muchísimas cosas funcionan así de mal.

Los responsables del "curso" son profesores de la Escuela de Ingenieros Aeronaúticos de la U.N.E.D., nada menos.

Mi uso del PC

Desde antes de comprarme el QL tenía claro que a mí me gustaban los ordenadores. El QL fue simplemente la confirmación. Es una herramienta magnífica y durante muchos años no he sentido la necesidad de tener otras herramientas informáticas, porque con él podía hacer todo lo que quería.

Sin embargo, desde hace unos años he sentido la necesidad de tener más posibilidades, además de las que me ofrecía el QL. Primero me compré un PC portátil para poder trabajar en cualquier sitio y después un PC de sobremesa para que diera cobijo a una QXL y también para enchufarle varios periféricos que me interesaban: digitalizador, tarjeta de sonido y lector de CD-ROM.

Cuando me compré el PC, sólo usaba MS-DOS como sistema operativo, y como es tan malo, era una tortura. Pero el compilador de C que usaba es muy bueno. Es decir, había cosas buenas y cosas malas.

En el PC de sobremesa empecé a usar Windows, que me dio bastantes errores al principio, pero me sirvió para conocer un programa maravilloso: PageMaker. Entonces pensé que si para usar PageMaker había que manejarse con Windows, merecía la pena. Las ventajas pesaban más que los inconvenientes.

Pasé por una fase en que usaba el QL y el PC casi simultáneamente para un mismo trabajo, cada uno en una fase. Era incómodo, de modo que empecé a ocuparme por encontrar buen software para PC que me sirviera para trabajar con comodidad sustituyendo las cosas que tan buen resultado me habían dado en el QL.

En el mundo PC hay muchos programas, y no siempre los mejores son los más conocidos, de modo que he pasado por varios errores hasta que he ido configurando mi sistema de una manera agradable.

Ahora utilizo exclusivamente el PC para todas las tareas que exijen sacar información al exterior, y el QL ha quedado para algunas cosas muy personales.

Por si a alguien le sirve, voy a explicar los programas que uso más habitualmente en el PC, comparándolos con los equivalentes del QL, cuando sea posible. Tened en cuenta que sólo explico los "éxitos", no los programas que he usado pero no me han servido.

* Editor de textos

  Evidentemente, ésta es una herramienta imprescindible. Estaba habituado en el QL a la flexibilidad de The Editor y a la velocidad de Arc. Por fin encontré el editor para MS-DOS "Vedit Plus", que es completamente configurable, puede manejar ficheros de texto y binarios, varios a la vez, y es programable
page 4
  mediante macros. Excelente, aunque su precio es de unas 40000 pesetas. Dentro de poco aparecerá una versión para Windows. La actual es para MS-DOS y sólo ocupa unos 100 Kb, es decir, es una miniatura para lo que se acostumbra en el PC.

* Entorno de desarrollo

  No he encontrado aún nada semejante a la facilidad del SuperBASIC, que combine programación y órdenes del sistema operativo. En otros sistemas operativos sí se que hay cosas semejantes: los scripts de UNIX y REXX en el OS/2, pero para MS-DOS o Windows no sé de nada semejante, así que me conformo con los ficheros BAT de MS-DOS, ayudados por las posibilidades del editor y completados por algunos programas que escribo con Olimpo.

* Programación

  Ya no echo de menos SuperBASIC para programar, ya que programo en C. Evidentemente, utilizo Olimpo para casi todo, de modo que tengo acceso a mis propias funciones. Como compilador de C uso Turbo C++ 1.0, que sólo ocupa unos 5 Mb y es rápido. Me resulta más cómodo que C68 porque da mensajes de error más claros y la documentación es muy completa.

* Fuentes de letras

  Este es uno de los motivos más importantes por los que uso el PC para todo lo

que tiene que salir al exterior. Utilizo fuentes TrueType. Las instalo en Windows y accedo a ellas desde todos los programas. Son escalables, se pueden modificar, puedo incluso crearlas, son de una calidad profesional y se encuentran con facilidad de muchos proveedores. En estos momentos tengo unas 1000 fuentes disponibles. En el QL no hay nada semejante. Antes me limitaba a las fuentes para mi DeskJet 500 que compraba en disquetes, pero no eran tan flexibles.

* Procesador de textos

Desde luego, la aplicación fundamental para casi todo el mundo. En el QL empecé como todos con Quill y luego me pasé a Perfection, pero son incomparables con el que uso ahora: Ami Pro 2.0. Es pequeño (a partir de 7 Mb), cómodo de usar, programable, permite dibujar dentro de él, tiene editor de ecuaciones matemáticas (fundamental para mi trabajo) y se puede personalizar fácilmente. Incluso es barato, aunque ya no se puede encontrar porque la versión actual es otra más avanzada y también más plasta, porque está llena de bobadas.

* Hoja de cálculo

Yo la uso mucho, porque pongo las notas con ella. Con Abacus me iba muy bien, apenas tenía problemas, pero ahora uso Excel 3.0, que es más cómodo en muchos aspectos. Al ser una versión antigua, no ocupa tanto como las actuales. Aún así, utiliza estilos y tiene macros. Es el primer programa de Microsoft con el que trabajo a gusto.

* Diseño gráfico

page 5

Ni siquiera el magnífico LineDesign puede compararse al programa que uso: Corel DRAW! 4. Este programa es extraordinario. Se puede conseguir en CD-ROM por 10000 pesetas y ofrece un sin fin de posibilidades. Incluye 19000 dibujos de calidad. Se puede hacer de todo, importar y exportar datos de muchos tipos e incluso incluye en el paquete un programa de dibujo de bitmaps. Evidentemente, Corel DRAW! es de dibujo vectorial. Yo lo utilizo mucho, para crear anuncios para las clases, tablas matemáticas, ilustraciones para mis apuntes, y también lo explico a mis alumnos de informática.

* Maquetación

Cuando hay que poner juntos varios artículos y dibujos, lo más lógico es usar un programa de maquetación. PageMaker sólo tiene un problema: cuesta algo más de 100000 pesetas. Eso sí, es absolutamente profesional. Hay que usarlo para darse cuenta de la potencia que encierra.

* Gestor de base de datos

No me gusta Archive, nunca me ha gustado; pero tengo varias bases de datos personales en su formato, que quiero convertir a formato dBase y usar en el PC. No he encontrado un gestor para PC que me convenza, pero me voy apañando con DBU, una sencilla utilidad que viene con código fuente con el compilador Clipper 5.01. También conozco, y me gusta, el gestor de bases de datos documentales Knosys (lo explico en el instituto). Con BC++ también viene una pequeña utilidad para Windows que maneja bases de datos dBase. Todavía no sé cuál usaré, pero pronto abandonaré Archive.

Como véis, cuando se habla de programas de usuario, las posibilidades que da un PC son muy interesantes. Y respecto a los sistemas operativos, recordad que en un PC se pueden montar varias versiones de UNIX, como por ejemplo Linux (que es gratis) y también el magnífico sistema de 32 bits OS/2. En fin, que el hardware del PC es barato, fiable, flexible y ya no es tan importante la dependencia que había antes respecto al MS-DOS.

Mi PC sólo tiene 4 Mb de RAM, y aún así trabajo con bastante comodidad. Es cuestión de saber hasta dónde se puede llegar.

Mi futuro

Ahora mismo no tengo dinero para invertir en informática, pero sí sé lo que quiero hacer cuando lo tenga. Me interesa comprar una impresora nueva. La DeskJet 500 que tengo da 300 puntos por pulgada, que para texto está muy bien, pero para imprimir fotos queda un poco pobre. Me gustaría comprar una láser de 600 puntos por pulgada, la HP LaserJet 5P.

Respecto a hardware para el ordenador, planeo comprar más RAM para el PC y quizá una unidad de almacenamiento Zip Omega, que me serviría para el PC de sobremesa, su QXL y el portátil. También estaría bien comprar un disco duro de un giga, que se puede conseguir por unas 40000 pesetas.

Respecto al sofware, en principio no hay nada por lo que tenga interés, quiero crear mis programas, para lo que recesito tiempo para estudiar el mucho

material que tengo: compresión de datos, manejo de gráficos, un pequeño intérprete, etc. Me interesan algunos programas de dominio público, especialmente, claro, C68.

Quiero programar para más entornos: Linux, MS Windows. Y llevar mis programas al QL por medio de Olimpo. Es decir: diversificar, personalizar y trasladar.

Me interesa que QLíper siga vivo y también publicar pequeños programas en otras revistas y distribuidores de dominio público.

A corto plazo, estoy trabajando en el programa que convierte ficheros de Archive a dBase y en una versión de Anarit para MS Windows.

El proyecto para Oxford University Press

En diciembre pasado se pusieron en contacto conmigo unos editores de esta empresa, que pretende comenzar a tomar posiciones en España, con la vista puesta en vender textos para los chavales que van a seguir los nuevos cursos de Primaria y Secundaria que se avecinan.

Me propusieron que les presentara un proyecto de cómo serían los libros que yo haría para los dos cursos del nuevo Bachillerato, de matemáticas. Se lo tenía que entregar a mediados de febrero. Formé un "equipo de trabajo" con un amigo con el que estudié la carrera, y ya hemos entregado el proyecto. Hemos usado Ami Pro para los textos y la maquetación final y Corel DRAW! para las ilustraciones.

Ahora estamos esperando a que en abril nos digan si les gusta nuestro proyecto o bien la van a encargar los libros a otro equipo. Si nos dicen que adelante, nos esperan dos años de mucho trabajo para sacar adelante los dos libros que habría que hacer.

Infovía

A mí todos estos asuntos del Internet y tal me dejan bastante frío. La gente con la que he hablado que realmente han estado conectados me han hablado de tiempos de espera de 10, 15, 20 minutos cada vez que acceden a un servidor. Claro, la mayoría de los servidores están en Estados Unidos y las comunicaciones no son rápidas.

Sin embargo, sí me resulta más atractiva la idea de una red española. Quizá Infovía se convierta en una alternativa fiable y rápida. Os pregunto, para el que lo sepa, ¿ofrecerá Infovía un servicio de correo electrónico por el precio de la conexión? ¿O habrá que pagarlo aparte? ¿O no habrá tal servicio?

Si se pudiera disponer de correo electrónico por el precio de una llamada local, me interesaría mucho conectarme a Infovía, para poder comunicarme y mandar ficheros a otros usuarios de España, vosotros los primeros. ¿Cómo lo véis?

Pedro Reina, D.25.2.1996

<<<< SPECTATOR, ZX SPECTRUM EMULATOR -- version 1.85 >>>>

by Dr. Carlo Delhez, November 1994

This file is in 80-column ASCII format. Use an ASCII editor to read the contents, import into your favourite word processor or copy directly to your printer (approximately 31 pages at 60 lines/page).

CONTENTS

MAIN FEATURES

Spectator is a no-nonsense ZX Spectrum emulator for Qdos-based systems. Here's an overview of its main features:

+ 48k or 128k ZX Spectrum emulation, either with or without a ZX Interface 1; the hardware mode can be changed at any time while running Spectator: you'll need just one executable file!
+ runs on the Super Gold Card and on the QXL, in both cases with the cache on; runs in any resolution of the QXL;
+ highly compatible emulation of the Z80 microprocessor, including all the non-official instructions; over 99% of existing Spectrum software will run on Spectator without any problems!
+ exact emulation of bank-switching in 128k hardware modes, including mirror addressing of RAM pages and double-screen mode; thus, all 128k programs will work!
+ written entirely in assembly language for compact, fast, reliable and efficient code;
+ the only Spectrum emulator for Qdos that supports multitasking with non-destructive windows (pointer interface needed); at ALL times, Spectator will automatically suspend itself when put into the background (by whatever method you prefer, e.g. by CTRL-C or by buttoning the job); neither screen nor sound nor keyboard will

page 8

interfere with other jobs;
+ two keyboard modes: one for serious applications, one for games;
+ three sound modes: sound via IPC beeper, sound via net-port, or sound off; sound via the IPC beeper will reproduce Spectrum frequencies independently of the emulation speed!
+ extensive yet easy-to-understand menu system which you can use either to taylor Spectator to suit your needs or for reaching the many special features offered by Spectator;
+ simple keypresses available for Z80 reset, Spectrum SuperBreak, screen colour inversion, graphical keyboard help screen, etc.; no need to enter the menu system for that!
+ three different types of snapshots (Z80, SNP, MBF) can be loaded; the widely available Z80-type snapshot files can be created; this allows easy exchange of Spectrum programs with users of the 'Z80' emulator for PC systems (and many other emulators as well);

+ Spectrum tape handling is possible in three different ways: SPT, TAP and TAP++; the latter copes with headerless files and allows direct exchange of tape files with the 'Z80' emulator for PC systems;
+ virtual microdrive emulation; Spectator was the first Spectrum emulator for any computer system worldwide with this feature; by now, the microdrive file format of Spectator has been accepted as a standard and is also supported by at least four (!) other emulators;
+ RS232 support, e.g. to let Spectator communicate with a real Spectrum so as to send programs or data hence and forth;
+ four widely used types of Spectrum joysticks can be emulated via the QL arrow keys, hence also via a QL joystick connected to CTL1;
+ screen output is exactly synchronized with Z80 interrupts; this mimics the way the Spectrum takes care of its screen refreshes;
+ dynamic screen refresh- and interrupt rate, automatically adjusted to the current emulation speed; this guarantees top-compatibility on the fast Qdos systems of the future;
+ two screendump options let you save all those beautiful Spectrum screens to disk for later use, e.g. in DTP software;
+ the main Spectator screen contains lots of useful information items, including a reliable speedometer, telling you the present emulation speed; but all this info can also be disabled, showing just the main Spectrum screen and nothing else;
+ two help screens are available: one with the Spectrum keyboard layout and one with a reminder of the various special keypresses used by Spectator;
+ finally: Spectator is a shareware product, made available for free to the whole Qdos community; registration is optional but highly appreciated. Registered users will get a beautiful laserprinted manual, a collection of disk conversion utilities (including Disciple), a free update, and info about later releases.

REGISTRATION

Spectator is distributed on a shareware basis. You are allowed to make copies for other Qdos users, provided you copy ALL the files and do NOT make changes in any one of the files.

If you like to use Spectator, please express your appreciation and
page 9
support my efforts by becoming a REGISTERED user. You will then receive a nicely printed copy of this manual, you will be informed about new releases, and can update to a more recent version for a small fee to cover my costs. Registration also includes free technical support by the author and the possibility to obtain a custom-made version of Spectator to suit your specific hardware configuration.

To become a registered Spectator user, send a EuroCheque worth NLG 60 (Dutch Guilders) or NLG 60 in cash, plus a note stating your name and address to me (my address can be found at the bottom of this document). The most recent version of Spectator on a 3.5" 720k floppy (3M-brand) and a laser-printed copy of the manual will be dispatched to you.

IMPORTANT: For cheques other than EuroCheques, please add NLG 20 for bank service costs. If you wish, you may also send cash (banknotes only!) in a currency other than Dutch Guilders, but please use an acceptable rate of exchange and add the equivalent of NLG 10 for my exchange costs. Thank you.

DUTCH USERS ONLY: Betaling van het registratie-bedrag vanuit Nederland kan ook via een giro-overschrijving. Stort het bedrag (60 gulden) op rekening 2438038 t.n.v. C. Delhez, Steenbergen, onder vermelding van 'Spectator registratie'. U krijgt het complete pakket binnen twee weken na ontvangst thuisgestuurd. Om zeker te zijn van een correcte (en vooral snellere) afhandeling, verdient het de aanbeveling mij ook middels een briefkaart, met vermelding van Uw naam en adres, over Uw bestelling te informeren.

STARTING SPECTATOR

For running Spectator, the following is required:

+ A Qdos-based system with at least 400k of free RAM memory (to store the program and to allocate the required work space); fast Qdos systems (such as Super Gold Card or QXL) are advised for best performance.

+ The Pointer Environment (i.e.: Ptr_Gen and WMan).

+ Toolkit-II.

Compatible hardware or software is ofcourse allowed as well. For technical reasons, Spectator never checks available memory and will crash sooner or later if not enough memory can be allocated.

Before starting Spectator, make sure Toolkit-II and the Pointer Environment are present (or create a simple boot-program to do this, see 'Spectator_Xample_boot'). Then type:

```
EX FLP1_SPECTATOR_EXE         ( or similar )
```

Simon Goodwin pointed out that SpeedScreen users should set MODE 8 before loading Spectator as to avoid problems with MODE calls that go

directly to the IPC.

First you will see a title-screen with some general information about Spectator. In the higher resolutions of the QXL, the Spectator window will be centered on the screen. Once the title-screen has disappeared, Spectator will perform a normal restart (RST 0). Eventually, the well-known Sinclair copyright message should appear in the bottom of the screen. Then Spectator is ready to be used.

Before typing anything, return to SuperBasic by pressing CTRL-C (or similar) and type the command JOBS. You will see that Spectator consists of three jobs, viz.:

+ Spectator : This is the main part of the emulator, responsible for the translation from Z80 to MC68000 code, generating of interrupts, screen-updates and keyboard reading.

+ Spc_Guard : This job continuously checks if Spc_Vita still exists and will immediately kill (= remove) Spectator if not.

+ Spc_Vita : The Spirit of Spectator; as soon as this job is removed, Spectator kills itself (more about that later).

The former two jobs have default priorities of 127 and 1, respectively. The priority of the latter job should always be 0. Just for fun, try:

```
SPJOB SPC_VITA,127
JOBS
```

Besides the above three jobs, also a 50 Hz interrupt has been linked to keep the flash frequency fixed at 1.5 Hz. Because of this interrupt, keep in mind NEVER to delete either the Spectator or the Spc_Guard jobs. The ONLY proper way to stop Spectator is

```
RJOB SPC_VITA         ( or equivalent )
```

Depending on the current keyboard interpreter setting, Spectator will either kill itself immediately, closing all channels, releasing its work-space and also unlinking the 50 Hz interrupt, or do this at the very moment it is forced to the foreground, e.g. by CTRL-C. Any other way of removing Spectator will eventually crash your QL!

In Pointer Interface terms, the Spectator job is completely "locked". So, when Spectator is temporarily "buried" by any other job, the Z80 program counter will be halted immediately, no more interrupts will be

generated, the screen won't be updated, and keyboard will no longer be read; i.e. Spectator is fully suspended and won't consume any CPU time. All these activities are immediately re-activated as soon as the Spectator window is on the foreground again. All this implies that neither the reading of the keyboard nor the printing of the screen nor the high-priority Z80 emulation will interfere with other jobs.

Having read all this, switch back to the Spectrum screen (by pressing CTRL-C or any equivalent method) and start having fun!

page 11

COMMANDSTRING QUALIFIERS

Instead of starting Spectator as described above, it is also possible to add a commandstring with one or more qualifiers so as to influence the initial settings of Spectator. A qualifier is a single-letter command, preceeded by any one of the characters '-', '+' or '/' (there is no effective difference between these three prefixes). The commandstring must be passed as follows:

```
EX SPECTATOR_EXE;<commandstring>        ( or similar )
```

with <commandstring> a character expression, normally a quoted string. The commandstring may contain one or more qualifiers, which may (but need not be) separated by one or more spaces. The Spectator commandstring interpreter is case-insensitive, so upper or lower case text may be used at will. Qualifiers are interpreted left to right, later settings superseding earlier settings.

Currently, the following qualifiers are supported:

```
/#        force keyrow mode
/Hn       select initial hardware mode:
           /h1 : 48k Spectrum
           /h2 : 48k Spectrum + ZX Interface 1 [default]
           /h4 : 128k Spectrum
           /h5 : 128k Spectrum + ZX Interface 1
/I        force inverted screen colours (EOR 7)
/Jn       select initial joystick type:
           /j0 : no joystick (keyboard) [default]
           /jC : Cursor joystick
           /jK : Kempston joystick
           /j1 : Sinclair ZX-IF-2, #1
           /j2 : Sinclair ZX-IF-2, #2
/K        force keyboard queue mode [default]
/N        force sound via QL network
/s        disable sound [default] (lowercase s!)
/S        force sound via IPC beeper (uppercase S!)
```

Example: EX SPECTATOR_EXE;'-h5+n/Jk'

starts a 128k Spectrum plus ZX Interface 1, generates sound via network and selects Kempston joystick

Future versions of Spectator will support many more commandstring qualifiers in order to to set things like default drives, RS232 redirection etc. Any invalid qualifier in the commandstring will result in an error message being printed in #0 or #1 and execution of Spectator will be aborted without any Qdos error.

SCREEN OUTPUT

page 12

The Spectrum has 50 screen-frames per second. Spectator emulates this to a very accurate degree by relating the number of screen-refreshes automatically and dynamically to the current emulation speed. If the emulation speed is 70%, then you'll get 35 screen updates per second,

but if (for some reason or another) the emulation speed increases to 130% some time later, Spectator will immediately respond and increment the rate to 65 screens per second.

Additionally, screen output is synchronized with generation of interrupts. The period inbetween two successive Z80 instructions is divided into four equal parts. In case of 'normal' synchronization, the screen will be updated at the end of the second part, i.e. exactly inbetween two Z80 interrupts. Synchronization can also be set 'low' or 'high' via the menu system (F1, option T, key Y); then, the screen will be updated at the end of the first or third part, respectively. The below diagram illustrates this:

```
     -----> time axis
'''''|'''''''''|'''''''''|'''''''''|'''''''''|'''''''''|'''''
     Z80       low       normal    high      Z80       etc.
     int       sync      sync      sync      int
```

If the sprites in a certain game are flickering a lot or seem to be invisible for most of the time, then try to change the synchronization. For example: in "BC's Quest for Tires", normal synchronization will show the cyclist on the screen without any flickering, but in case of low or high synchronization it disappears alltogether! In fact, normal synchronization will be the best choice for most programs, yet it's nice to have some control of the synchronization anyway.

The ZX Spectrum FLASH attribute is emulated by software, as the QL hardware flash works in a completely different fashion. Spectator fixes the flash frequency to 1.5 Hz, but if the priority of Spectator is too low, a frequency smaller than this value is used (in that case you get one flash-flip per sceen update). The ZX Spectrum BRIGHT attribute is neglected as the QL has only 8 colours. Only in rare cases, this causes information on the screen to be invisible.

KEYBOARD CONSIDERATIONS

Spectrum BASIC is tokenized. This is handy in the sense that you need less keystrokes for often used commands like 'PRINT': you merely have to press 'P' when the cursor is a 'K' and the whole keyword appears. A disadvantage is the fact there are no less than 91 tokens and these are all hidden under no more than 40 keys. These 40 keys also have to provide the regular ASCII set (96 characters), special control characters (25 in total) and a set of (user defined) graphics (37). This makes a a grand total of 249. So, each of the forty keys can be used for an average of 6.2 functions (some keys have only one function, many have 7, some 8 or even 9).

Obviously, it is quite impossible to know by heart where all these
page 13
functions are located. The original Spectrum keyboard has the functions printed on the keyboard itself. Naturally, the QL does not provide this facility; therefore, a help-screen is available which is a graphical image of the Spectrum 48k keyboard. Press CTRL-F1 to pop-up the help screen. Press ESC to return to the Spectrum screen. It is not the purpose of this manual to explain how to use the Spectrum keyboard. Please consult a Spectrum handbook to learn this.

As said before, the 48k Spectrum keyboard has only 40 keys. All of these keys are also available on the QL:

+ 26 letter keys (A thru Z)
+ 10 number keys (0 thru 9)
+ 1 Space key (Space bar on the QL)
+ 1 Enter key (same on the QL)
+ 1 Capital-Shift key (Shift on the QL)
+ 1 Symbol-Shift key (CTRL on the QL)

It is very important to note the difference between CAPITAL SHIFT (the QL SHIFT key) and SYMBOL SHIFT (the QL CTRL key)!

The related 40 keys on the QL should be sufficient to operate Spectator. However, there are some strange incompatibilities. Just some examples:

To get an uppercase letter on the QL, one presses Shift plus the letter key (e.g. Shift-'p' gives 'P'); on the Spectrum one has to press Capital-Shift plus the letter key, i.e. just the same. But: to get the symbols of the numeric keys of the QL, one also needs to press Shift plus the number key (e.g. Shift-'4' gives '$'). However, on the Spectrum one has to press Symbol-Shift (NOT Capital-Shift!) plus the number key, so this means: you have to press CTRL-'4' on the QL to get the dollar. Pressing SHIFT-'4' on the QL (i.e. Capital-Shift '4') produces 'Inverse Video Mode'.

Having read the above example, one would press CTRL-'8' to get an asterisk (the '*' symbol). Strangly enough, this keystroke produces the opening bracket '(': the symbols on the Spectrum keyboard are arranged in a strange non-standard fashion:

key:	1	2	3	4	5	6	7	8	9	0
CTRL-key:	!	@	#	$	%	&	'	(	)	_

The asterisk is 'hidden' under CTRL-B and the '^' (hat) symbol can be typed as CTRL-H. Other symbols are distributed over various letter keys:

key:	B	C	H	J	K	L	M	N	O	P	R	T	V	X	Z
CTRL-key:	*	?	^	-	+	=	.	,	;	"	<	>	/	Pound	:

CTRL plus any one of the remaining letter keys produce a special token (e.g. CTRL-A gives 'STOP').

page 14

It is obvious that it is not quite easy to use the QL keyboard as a Spectrum typewriter. An additional problem is the fact that the Spectrum does not have a 'type ahead buffer' (keyboard queue). So, the keyboard is not scanned until a previous keypress has been fully processed. On the normal Spectrum this is not really a problem, unless you are editting very long lines of BASIC. On Spectator, this could be a distinct problem if it is running slower than the Spectrum (Trump Card or Gold Card). Using the QL keyboard queue might be a solution, but then many games don't work since these allow you to press many keys at once for complicated steering mechanisms; these multiple keypresses are ignored by the QL keyboard queue handler. Additionally, pressing down a single key continuously cannot be 'seen' through the queue handler, as it starts to auto-repeat with a finite repetition frequency.

As a solution to all this, Spectator provides TWO keyboard emulation modes:

(1) Keyboard-queue-mode, indicated by a 'K' in the top-righthand corner (default when Spectator is started)

Advantages:

+ Type-ahead buffer, so you don't have to wait until a keypress has been processed
+ Excellent keyboard mapping: all ASCII symbols on the QL keyboard can be typed (just using Shift, not CTRL)
+ All-time autorepeat (origin: QL keyboard handler)
+ No unwanted keyboard input when job is buried
+ Very well suited for serious applications

Disadvantages:

- Some tokens less easily accessible

- Limited emulation of Spectrum 'keyrow' map
- No emulation of original Spectrum key repetition
- Delayed recognition of special CTRL keystrokes
- No joystick emulation possible
- Not suited for playing games

(2) Keyrow-mode (indicated by a '#' in the top-righthand corner)

Advantages:

+ Excellent emulation of Spectrum keyrow map
+ Original Spectrum keyrepeat emulated correctly
+ Tokens more easily accessible
+ Immediate recognition of special CTRL keystrokes
+ Joystick emulation via cursor keys and space bar
+ Very well suited for playing games

Disadvantages:

page 15

- No keyboard queue, results in dull typing
- ASCII symbols not easily accessible
- Only auto-repeats when the Spectrum ROM code is used
- May 'see' unwanted keystrokes while the job is buried
- Not suited for serious applications

You can switch from one mode to another by typing CTRL-F3. The symbol 'K' changes into '#' or vice versa. Alternatively, you may use the pop-up menu system (F1, option T, key K).

Ad (1) : KEYBOARD-QUEUE-MODE

All the letters and symbols can be typed on the QL keyboard as though you were using a QL editor. You don't have to use CTRL: both capitals and symbols on the number keys can be obtained by pressing the Shifted key.

Dedicated symbol keys (like '.', '/' and ';') can be used as on the QL (also Shifted). There are a few exceptions: the keys '[', ']', '{', '}', '\', '|', '~' and '(copyright)' can only typed when the Spectrum is in extended mode (E-cursor). First put the Spectrum in extended mode by pressing F5 and then press the appropriate key(s) for the symbol you need.

arrows	work just fine
Shift-Space	is break
CTRL-left	produces 'delete' (equivalent to CTRL-0)
CTRL-Capslock	switches Spectrum CAPSLOCK mode but does NOT alter QL CAPSLOCK (equivalent to CTRL-2); take care NOT to switch on QL CAPSLOCK as this will disable a large number of the keyboard functions
F4	produces 'edit' (equivalent to CTRL-1)
F5	flips extended mode (equivalent to pressing CTRL and SHIFT together, but that is not seen through the queue handler!).

Further examples showing how to obtain tokens and special functions:
(please take a look at the help screen for more information!)

cursor	keypress	result	notes
K	Q	PLOT	
K	Shift-Q	PLOT	
K/L	CTRL-Q	<=	CTRL-key works ONLY for tokens
L	Q	q	
L	Shift-Q	Q	
E	Q	SIN	
E	Shift-Q	ASN	don't use CTRL-Q
G	Q	udg-Q	
G	Shift-Q	udg-Q	udg = user definable graphic

K/L	1	1	
K/L	Shift-1	!	
K/L	CTRL-1	edit	
			page 16
E	1	blue paper	
E	CTRL-1	blue ink	
E	Shift-1	DEF FN	
G	1	graphic	
G	CTRL-1	inverse graphic	
G	Shift-1	inverse graphic	

The keywords of the numeric keys '6' thru '0' need some further explanation. Similarly to Shift-'1' in extended mode for 'DEF FN', you are tempted to press Shift-'9' in extended mode for 'CAT'; this however is not right: Shift-'9' is converted to SymbolShift-'8' for the sake of the opening bracket '(', so you get 'POINT'. To get 'CAT', press Shift-'0' in extended mode, as this is converted to SymbolShift-'9'. It is good to remember that the RED symbols on each Spectrum key are also on the SAME QL key. You therefore have to press the single quote key in extended mde to get the 'ERASE' command, while the underscore key ('_') is used for 'FORMAT'.

Finally a WARNING: ALL keypresses are seen and queued. If you're the type of person that frantically hits all the keys of the keyboard when a program doesn't react quickly enough to your commands, keep in mind that ALL these useless keypresses are remembered and executed, sometimes with disasterous results...

Unfortunately, the queue-facility of K-mode is lost in the '128 editor; still, all the other features of K-mode remain the same.

Ad (2) : KEYROW-MODE

All the letters and symbols must be typed as though you were using a Spectrum keyboard. Use Shifted letter keys for uppercase, but press CTRL plus a numeric key to get the symbol on that key.

Dedicated symbol keys (like '.', '/' and ';') can be used as on the QL, but you cannot Shift them (for example: Shift-'/' still produces '/' instead of '?') There is no special provision for extended-mode symbols.

arrows	work just fine
Shift-Space	is break
CTRL-Left	produces 'delete' (equivalent to Shift-0)
Capslock	switches Spectrum CAPSLOCK mode but ALSO QL CAPSLOCK mode (it is better to use Shift-2)
F4	produces 'edit' (equivalent to Shift-1)
F5	flips extended mode (equivalent to pressing CTRL and SHIFT together).

Further examples showing how to obtain tokens and special functions:
(please take a look at the help screen for more information!)

cursor	keypress	result	notes
			page 17
K	Q	PLOT	
K	Shift-Q	PLOT	
K/L	CTRL-Q	<=	CTRL-key works also for symbols
L	Q	q	
L	Shift-Q	Q	
E	Q	SIN	
E	Shift-Q	ASN	advice: don't use CTRL-Q
G	Q	udg-Q	
G	Shift-Q	udg-Q	udg = user definable graphic

```
     K/L       1         1
     K/L       Shift-1   edit        \
     K/L       CTRL-1    !           |   Note the differences with
     E         1         blue paper  >   respect to the Keyboard-
     E         CTRL-1    DEF FN      |   Queue-Mode
     E         Shift-1   blue ink    /
     G         1         graphic
     G         CTRL-1    inverse graphic
     G         Shift-1   inverse graphic
-----------------------------------------------------------------------
```

To get the '?' symbol you would need to press CTRL-C. This is not possible as this keypress is intercepted by the job manager. The backslash key '\' produces a question mark without switching jobs. For the backslash itself, press Shift-'D' in extended mode. Using CTRL-C for switching jobs MAY be seen by Spectator and causes a repetition of '?' characters in the period that Spectator remains buried; this cannot be avoided.

None of the keywords or functions are converted to deviating key presses, so you can directly use all the keys shown on the help screen.

There is no queue, so type slowly and firmly. It may be good practice to change the Spectrum repetition system variables for better response (REPDEL and REPPER are at $5C09 and $5C0A respectively).

KEYBOARD CONTROL

There are several special Control keypresses. These are not available on the normal Spectrum but may be quite useful for the emulator. They are summarized below:

F1 — Pop-up Spectator main menu; see section 'Pop-Up Menu System'.

CTRL-F1 — Show keyboard Help screen; return with ESC. The help screen can also be seen via the pop-up menu system (F1, option H).

CTRL-F2 — Negative screen colours (EOR 7); press CTRL-F2 again to return to positive colours. The screen colour can also be changed via the menu system (F1, option T, key C).

CTRL-F3 — Toggle keyboard mode (Keyboard-Queue-Mode is indicated by a 'K', Keyrow-Mode by a '#', see previous section). The keyboard mode can also be changed via the pop-up menu system (F1, option T, key K).

CTRL-ALT-F3 — Toggle sound; sound off is indicated by 's' (lower case), sound via IPC beeper on is indicated by 'S' (upper case) and sound via network port by 'N'. See section 'Beeping' for more details. The sound mode can also be changed via the pop-up menu system (F1, option T, key S).

CTRL-ESC — Refresh Spectrum ROM and force a Break in the current program (also in machine code); this may crash your Spectrum (in case of a crash: press CTRL-ALT-ESC to recover).

CTRL-ALT-ESC — Refresh Spectrum ROM and perform a RST 0 (Z80 reset) (the safest way to stop protected games or to recover from a crash; works like the Non-Maskable Interrupt on a Spectrum). This also sets the keyboard mode to 'K' (queue mode) as this mode seems most suited for Spectrum BASIC. A reset can also be accomplished via the pop-up menu system (F1, options R or X).

In Keyboard Queue Mode, a CTRL-keypress is stored in the queue and is not seen until the Spectrum ROM or a Spectrum program reads the keyboard. This may cause a delayed recognition of the keypress. In case of a crash, the keypress may not be seen at all. Remember that pressing the key more than once does not cure this! In Keyrow Mode, the CTRL-keypresses are always seen immediately, whatever the Spectrum is doing at that moment. Pressing CTRL-ESC in the Spectrum editor won't do any harm, but the effect in other editors is unpredictable.

POP-UP MENU SYSTEM

To make life for Spectator users even easier, a pop-up menu system has been implemented. The system of menus offers many facilities by pressing down just a few keys. The main menu can be popped up by pressing F1 while the emulator is running. A list with the available options is shown. In order to execute any one of the options, simply press the key printed in front of the desired option. Depending on the chosen option, either new menus are popped up (the '>' sign after the letter indicates this) or the required action is performed immediately (indicated by a '-' sign).

The main menu offers the following facilities:

B - back to Spectrum; equivalent to pressing ESC.

C > change defaults; a sub-menu appears with a list of defaults that can be altered. This sub-menu will re-appear after you have changed any one of the defaults; this makes it easier to change

various defaults at once; press ESC to return to the main menu. The sub-menu allows you to change the following things:

```
M  >  1st microdrive device (drives 1 thru 4) : <mdv1.dev>
N  >  2nd microdrive device (drives 5 thru 8) : <mdv2.dev>
T  >  SPT and TAP tapefile device             : <tape.dev>
U  >  TAP++ filename, used in TAP++ tape mode
Z  >  Z80, MBF and SNP snapshot device        : <snap.dev>
*  >  ReSave Spectator
```

The new device names may be any valid Qdos devices (including for example 'WIN1_', 'N1_FLP4_' or 'ATR2_') plus an arbitrary subdirectory path (probably most useful for Harddisk). The existence of the devices and subdirectories is NOT checked by Spectator. An underscore at the end of a device name is optional; Spectator will add one if you omit it.

After having set the various devicenames to suit your needs, use the 'ReSave' option to save a personally customized version of Spectator to disk. All the current settings of Spectator are effectively saved in such a customized version. This includes for example:

+ the hardware mode (48k or 128k, with or without ZX-IF-1)
+ the current keyboard mode ('K' or '#')
+ normal or inverse screencolour setting
+ the soundmode ('s', 'S', or 'N')
+ the current joystick setting
+ the default microdrive-, tape- and snapshot-devices
+ the various filenames used within the menu system
+ ...and much more!

This means that you don't need to set all these things again the next time you use Spectator! Note that the ReSaved version itself cannot be ReSaved again. This is to prevent distribution of unofficial versions of Spectator. If your version of Spectator has no ReSave option, it's an unofficial ReSave! It cannot be guaranteed that a ReSaved version will work in exactly the same way as the original; if it doesn't, delete the ReSaved copy and retry with the original version.

```
D  >  make screendump of Spectrum screen as QL MODE-8 file (filename is
      requested; overwrite confirmation and full error trapping); this
      screendump may be reloaded from SuperBASIC by a command like:

               LBYTES flp1_name_scr,SCR_BASE    ( or similar )

E  >  make screendump of Spectrum screen as Spectrum SCREEN$ file (SPT
      type only; filename is requested; overwrite confirmation and full
      error trapping); this screendump can be reloaded from Spectrum
      Basic by a command like:

               LOAD "name"SCREEN$            ( or similar )

                                   page 20
H  -  pop up the keyboard help screen (return by ESC);
I  -  show general program & author info (return by ESC);
J  >  joystick selection; a sub-menu appears, showing five possible
      choices:

      0  -  No joystick (keyboard option)
      C  -  Cursor, Protek or AGF joystick
      K  -  Kempston Joystick
      1  -  Sinclair ZX Interface 2, lefthand side
      2  -  Sinclair ZX Interface 2, righthand side

      Select the required option by pressing the appropriate key or
      using the up/down arrow keys. The current choice is highlighted.
      Activate your choice by pressing Enter or Space. Pressing 'R' or
      ESC will leave the old setting unchanged. See section 'Joysticks'
      for more information on joystick emulation.

K  -  present overview of main keyboard functions (return by ESC);
Q  >  quit Spectator (confirmation requested);
R  -  reset Z80 (hence Spectrum, NO confirmation requested!);
S  >  save memory contents as a new-style compressed Z80 snapshot
      (filename is requested; overwrite confirmation & full error
      trapping); the shapshot can be reloaded from Spectrum Basic by a
      command like:

               LOAD "/name"                  ( or similar )

      The new-style Z80 files cannot be reloaded on old versions of
      Spectator (1.21 or earlier).

T  >  change emulator settings; a sub-mneu with various emulator
      settings appears. Press the key in front of an option to change
      the setting. The menu will be updated immediately to show the new
      setting. Press ESC to return to the Spectrum. The settings you
      can alter are:

      2  -  issue-2 emulation: off or on (also see section "Z80
            compressed file support")
      C  -  screen colour: positive or negative (also see section
            "Keyboard Control")
      I  -  simple screen (without the information bars): on or off
            (also see section "On-Screen Information Bars")
      K  -  keyboard mode: queue or keyrow (also see section "Keyboard
            Considerations")
      S  -  sound mode: off, beeper or net (also see section "Beeping")
      T  -  tapefile type: SPT, TAP or TAP++ (also see section "File
            Management")
      Y  -  screen output synchronization: normal, low or high (also
            see section "Screen Output")

X  >  Spectrum hardware selection; a sub-menu appears, showing four
      possible choices:

      1  -  Standard 48k Spectrum
                                   page 21
```

2 - 48k Spectrum with ZX Interface 1 (default)
4 - 128k Spectrum
5 - 128k Spectrum with ZX Interface 1

Select the required option by pressing the appropriate key or using the up/down arrow keys. The current choice is highlighted. Activate your choice by pressing Enter or Space: note that this will automatically reset the Spectrum!! Pressing 'R' or ESC will leave the old setting unchanged and returns to the main menu. See section 'Hardware Modes' for more information on these options.

In general, the menus disappear by pressing ESC. When a text needs to be editted (e.g. a filename), ESC won't always work (depending on the details of your computer system); in that case, press the up or down cursor key instead.

128K SPECTRUM

Spectator can emulate both a standard 48k Spectrum as well as a 128k Spectrum. Emulation of the Sinclair ZX Interface 1 is optional in both cases. The hardware emulation is dynamic, i.e. it can be changed at any time. By default, a 48k Spectrum with ZX-IF-1 is used. The hardware mode can be changed via menu option 'X' (see section 'pop-up menu'). If a new hardware mode is selected, the Spectrum will automatically be reset.

The Spectrum 128 offers the following new features w.r.t. the 48k Spectrum:

(1) 32k of ROM: 16k editor ROM plus 16k main ROM, the latter being almost identical to the 'old' 48k ROM. Only one 16k ROM can be present at any time; bankswitching is used to select the required ROM. New software features (all properly emulated by Spectator) include

(a) menu system, moveable highlighted bar type;
(b) tape tester, tape loader, separate calculator;
(c) full-screen Basic editor with renumber option;
(d) extended Basic, e.g. to control '128 sound;
(e) 80k RAMdisk (e.g. accessed via SAVE !"TEST" or CAT!);
(f) protected 48k emulation mode.

(2) 128k of RAM: only 48k can be 'seen' at any time in the 64k memory map. The remaining 80k consists of five blocks of 16k each, which can be paged in in the C000-FFFF memory area via bankswitching and are used by the Editor ROM as a RAMdisk.

(3) Double Screen: two screenpages are available. The 'active' page can be selected via the bankswitch register and needn't even be present in the main 48k RAM memory area!

(3) High-quality soundchip: 3-channel sound with independently selectable volume, frequency, decay/attack mode and noise level.
page 22
This soundchip output is not emulated by Spectator as the QL lacks the required hardware. However, Spectator does keep track of OUTs heading to this chip, mainly for proper Z80 snapshot saving.

(4) RS232 and Midi Interface, not yet emulated by Spectator.

(5) protected 48k 'emulation' mode: this mode can be entered via the main menu of the 128k ('48k Basic') or by typing 'SPECTRUM' in the 128k editor. The latter method preserves the current Basic program. The former method resets the Spectrum before entering 48k mode. In both cases, the specific features of the '128 (like bankswitching) are no longer available. Use a hardware reset (e.g. CTRL-ALT-ESC) to return to '128k mode.

Please consult a Spectrum 128 handbook to learn more about its features

and about the way it is to be operated.

On the Spectrum, bankswitching (ROM and RAM) is done via hardware (controlled by OUT instructions), hence very fast. On Spectator, bankswitching can only be emulated via software and is (in comparison) very slow. For example, the Edit ROM calls the Main ROM very often, to do things like printing, drawing and calculating. These calls are performed via bankswitching (remember that only one ROM can be paged in at any time). This makes operation of the '128 editor very slow. Nonetheless, it does work and that's what counts!

There are some peculiarities of the bankswiching: for example, some pages may be doubly addressed in the main memory map, i.e. the same physical page is present at two different logical locations. Changing the page at one place will thus change it at the other place as well. Spectator supports these special features to a highly accurate degree (all 128k-snapshots I have available work OK on Spectator!). Try the following example and see for yourself:

(1) press F1, then 'X', select a 128k mode, and press Enter;
(2) when the 128 main menu is on-screen, select '128k BASIC' (using the arrow keys, and pressing Enter);
(3) type USR 0 to go to 48k mode without disabling bankswitching;
(4) type CLEAR 49000 to set the machine stack below $C000;
(5) type OUT 32765,21 to let the screenpage at $4000 be co-addressed at $C000 (i.e.: the contents of the $4000-$7FFF area are always equal to the contents of the $C000-$FFFF area);
(6) type POKE 49152,-1 : the screenbyte at 16384 will change as well!
(7) type POKE 16384,0 : the contents of address 49152 change as well!

ON-SCREEN INFORMATION BARS

In the top and bottom of the screen, you will see blue information bars that contain some relevant information about the current memorypage status and speed of Spectator. The various items are schematically reprinted below and have the following meaning:

+ Information bar at the top of the screen:

page 23

```
MAIN if-1 edit    | | |-| | |_| |~|     sK  x.xx
^^^^ ^^^^ ^^^^     ^ ^ ^ ^ ^ ^ ^ ^      ^^  ^^^^
the current ROM    0 1 2 3 4 5 6 7      soundmode, keymode
is highlighted     RAMbank setting      and Spectator version
```

Only one ROM name should be highlighted at any time, corresponding to the ROM currently being paged in:

```
main  :  the main 48k ROM
if-1  :  the ZX-Interface-1 ROM
edit  :  the edit ROM of the Spectrum 128k
```

There are 8 RAM pages of 16k each, numbered 0 through 7, and their distribution in the main 48k RAM of the Spectrum is shown by way of the small, horizontal bars. The above example must be interpreted as follows:

```
          page 5 is in the 4000-7FFF area,
          page 2 is in the 8000-BFFF area,
      and page 7 is in the C000-FFFF area.
```

The remaining pages (0,1,3,4,6) are stored outside the Z80 memory map. If you do not understand all this - don't worry. If you do, it may give you a good idea of how Spectrum 128 programs are making use of the bankswitching and how Spectator emulates some of the peculiarities. It must be noted that the info bar is updated simultaneously with the main Spectrum screen and may not always show the actual page distribution at a given moment in time.

+ Information bar in the bottom of the screen:

This contains a speedometer that indicates the instantaneous emulation speed of Spectator. The outer left side represents 0% and the outer right side represents 150%. Three specific positions are highlighted:

```
at   8% : approximate speed on Trump Card
at  32% : approximate speed on Gold Card
at 100% : original ZX Spectrum speed
at 130% : approximate speed on QXL and Super Gold Card
```

As you will see, the emulation speed will vary quite a lot around a certain equilibrium value. Some programs will push the speed bar far to the right, others will drag it back towards the left. Spectator is very honest about itself: you can safely trust the readout of the speedometer!

Although the information bars can be very helpful to determine what Spectator is doing and to see how some of the most important modes (like keyboard) are set, it may sometimes be more comfortable to see just the plain Spectrum screen -- without all the info. This can be accomplished via the pop-up menu system: press F1, select 'T' (change settings) and use option 'I' to enable (or disable) the 'simple screen'.

page 24

BUTTONING

If you put Spectator in a button, it will automatically suspend itself and won't consume any CPU time until you unbutton it again. The program you were running (e.g. a game) is halted in the most effective and reliable way, so after unbuttoning, this program is resumed as if it were never interrupted.

BEEPING

By default, Spectator produces no sound (lower case 's' in the top-righthand corner of the screen). Sound can be generated in two ways: via the internal IPC beeper or via the net port. Pressing CTRL-ALT-F3 will change the lower case 's' into an upper case 'S', indicating sound generation via the IPC beeper. Pressing CTRL-ALT-F3 once more, will display the letter 'N' to indicate sound generation via the net port. Press CTRL-ALT-F3 once more to disable sound again. The sound setting may also be changed via the pop-up menu system (F1, option T, key S). Note that the 3-channel output of '128 soundchip is NEVER emulated (but Spectator does keep track of its status!).

Although the generated sounds may sometimes be a real pain in the ear, it can also be quite advantageous to enable sound. Some games seem to have crashed on a mute system, while they are actually 'amusing' the player with some lengthy melody before commencing. In such a case, 'sound on' can be helpful for determining what the program is actually doing.

(1) 'S' ... sound via IPC beeper

The Spectrum can produce sound by regularly activating and deactivating the electromagnet in the loudspeaker. On the QL, sound is produced in a similar way, but it is fully controlled by the IPC (or by an 'emulated' IPC on a QXL) and therefore unaccessible directly from MC68000 machine code. Nevertheless, Spectator does try to mimic the Spectrum sound. For this, it registers the output heading for the Spectrum speaker during a period of 20 ms, and then sends a command to the IPC to generate a tone which resembles the tone originally produced by the Spectrum. As this involves an averaging procedure, quick variations occuring within the 20 ms period cannot be noticed and are lost in the produced sound. Nevertheless: Spectator will always be able to determine the EXACT frequency of tones, irrespective of emulation speed; therefore,

frequencies are reproduced properly on any Qdos system!

(2) 'N' ... sound via Net port

Much better sound can be generated via the QL net port. The way this port is accessed by the QL is very similar to the way the Spectrum accesses its internal loudspeaker. Therefore, the output of the net port will sound just the same as on the Spectrum, without loss of quality. There's just one draw-back: Spectator can

page 25

only control the net port when the Z80-emulation job ('Spectator') is running. Although this job has a very high priority, the scheduler will need to switch to other jobs once every while. When this happens, the sound stops abruptly and is only resumed when 'Spectator' gets CPU access again from the scheduler. Due to these silent periods, however short they may be, the output sounds fairly stammering and faltering. Obviously, frequencies are not reproduced but lessened or increased proportionally to the operating speed of Spectator with respect to the real Spectrum. It is my experience that this aspect has little influence on the quality of the sound.

In order to hear sound via the net port, you must connect it to your audio set (amplifier). There are several ways you can do this:

(a) Connect a small loudspeaker (or set of headphones) directly to the net port. This is a simple but somewhat blunt solution. It may blow-up your net circuit due to the large currents flowing.

(b) Connect the net port to the microphone input of your amplifier directly. This should do no harm to your QL net circuit.

(c) The same as (b), but with an additional low-pass or middle-pass RC-filter inbetween QL and amplifier in order to smoothen the sharp edges of the pulses produced in the net circuit.

In case you didn't know it already: the net ports are on the outer left when facing the back of your QL; they are marked 'NET' (how appropriate!). On the QXL it are the (only) two sockets on the back. It doesn't matter which of both ports you use.

INTERRUPTS

The ZX Spectrum usually operates in Interrupt Mode 1 (IM 1). The Maskable Interrupt Server at address $0038 is called 50 times per second to update the internal clock (Frames) and to read the keyboard. When Spectator would have to do this in the same way, execution speed would go down rather drastically, so IM 1 is neglected. Instead, Spectator updates the clock and reads the keyboard in fast 68000-code at a frequency that is automatically matched to the instantaneous emulation speed: if Spectator runs at 30%, 15 Hz interrupts are generated, and if Spectator runs at 130%, 65 Hz interrupts are generated! This results in top compatibility!

Many programs enable Interrupt Mode 2 (IM 2) which allows for user interrupt routines. In this mode, the interrupt address is fetched from the address whose high part is formed by the I register and whose low part is read from the data line (always $FF in Spectator). Ofcourse, Spectator supports this type of interrupt, and (as with IM 1) at a frequency matched to the instantaneous emulation speed. An IM-2 server may end its execution with a jump to address $38 so that the timer and keyboard are updated. This may conflict with the regular keyboard reading, leading to less accurate keyboard response. Programs suffering from this effect are (for example) Beta Basic and PopKey.

page 26

Interrupt Mode 0 (IM 0, default after reset but hardly ever used) executes the instruction on the data bus. On Spectator this will always be $FF (i.e. RST 38h), so IM 0 is just equal to IM 1 and is emulated as such (see above).

The Non-Maskable Interrupt (high-priority call to address $0066) is not supported by Spectator because no Spectrum program will use this, so the second interrupt flipflop (IFF2) is always equal to IFF1 (0 or 1).

The instructions EI and DI (enable/disable interrupts) function as expected, and, just as on the real Z80, the former only takes effect AFTER the next instruction has been executed. The instructions 'LD A,I' and 'LD A,R' will properly set the parity/overflow flag to the value of IFF2 (=IFF1!). HALT waits for an interrupt; on Spectator it operates in the same way in all three interrupt modes: it waits for the next interrupt, which is exucted and RETurns to the byte after the HALT instruction. Timing of HALT is as realistic as possible and reflects the current emulation speed. The instructions 'RETI' and 'RETN' are emulated as a regular 'RET' (but they are used next to never, even in the Spectrum ROM!).

ZX INTERFACE 1

Spectator supports both the software and the hardware of ZX Interface 1. When the Z80 interpreter reaches the byte $2A at address $0008 or the byte $23 at address $1708, the Interface 1 Shadow ROM (edition 2) is paged in (in the first 8k of memory only) and the Z80 interpreter continues emulation in this ROM. When the Z80 interpreter reaches the byte $C9 at address $0700, the original contents of the first 8k of the Spectrum ROM are restored and the emulator resumes execution of the Spectrum ROM. You can verify the presence of the ZX Interface 1 Shadow ROM by typing the command 'RANDOMIZE USR 8' which should return 'Hook code error' (a Spectrum without Interface 1 would return the cryptic '5 M' message). The main hardware components contained within ZX Interface 1 are:

```
(1) 8k Shadow ROM BASIC Extension   : see above
(2) A Microdrive Controller         : see section 'File Management'
(3) An RS232 Interface              : see section 'RS232 Serial Port'
(4) A Local Area Network connection : not yet implemented on Spectator
```

Application programs accessing these hardware components by way of the standard Hook Codes of the ZX Interface 1 Shadow ROM will work just fine. However, programs using their own INs and OUTs to achieve the same thing will most certainly fail to operate...

FILE MANAGEMENT

On the 'real' ZX Spectrum, there are several ways for storing and retrieving your data. First of all, the regular SAVE and LOAD commands with a quoted name which write to and read from a tape recorder.

page 27

Secondly, if you have installed ZX Interface 1 with one or more Microdrives, you can use the more advanced SAVE *"m" and LOAD *"m" commands which access the microdrives. Thirdly, several diskinterfaces exist which all have their own specific commands for accessing the drives.

Spectator supports all of the above possibilities (and more!).

(1) SPT tapefile support

The SAVE commands with just a quoted name will write data directly to a QL disk. The header (17 bytes) and the actual data block are stored together in a single file which is called

```
<tape.dev>_SPT_<n>_<name>      { SPT meaning: SPectrum Tape }

with       <tape.dev> : the tape device (FLP1 by default)
           <n>        : the filetype (0, 1, 2 or 3)
           <name>     : the name supplied in the SAVE command
                        (following ZX Spectrum standards, the name
```

is 1..10 characters long)

The filetype represents the type of data contained within the file:

```
0 : BASIC lines and variables
1 : numerical array
2 : string array
3 : code block (including screen$)
```

```
Examples:  SAVE "Test"           creates <tape.dev>_SPT_0_Test
           SAVE "Test2" LINE 0           <tape.dev>_SPT_0_Test2
           SAVE "Income" DATA x()        <tape.dev>_SPT_1_Income
           SAVE "DBase" DATA d$()        <tape.dev>_SPT_2_DBase
           SAVE "Pattern"SCREEN$         <tape.dev>_SPT_3_Pattern
           SAVE "Tables"CODE 65000,512   <tape.dev>_SPT_3_Tables
```

Since the filetype is part of the name of the produced file, it is possible to give a basic file (type 0) and a code file (type 3) the same name, although this is not advised.

The tape device <tape.dev> can be changed via the menu system. First select option 'C' (change defaults) and then option 'T' (tape device). You can edit the current setting and accept it by Enter. Pressing ESC (or cursor up/down) will leave the original devicename unchanged. Spectator will not check the existence of the device you enter.

It is obvious that files are stored more reliably on disk than on tape. Nonetheless, the VERIFY command is supported by Spectator, as you may want to compare a file on disk with resident data. Just as on the normal ZX Spectrum, VERIFYing a 'SCREEN$' will always fail due to the messages printed to the screen. Also, blocks of unequal length fail to verify. The message 'TAPE LOADING ERROR' is given in all these cases.

page 28

LOADing a file must be done by supplying the required name. LOAD uses the same filename convention as SAVE. The frequently used command LOAD "" (i.e. no name, load first program on tape) does not work and returns a TAPE LOADING ERROR. The same happens when you try to LOAD a file which doesn't exist.

```
Examples:  LOAD ""               loads   ...TAPE LOADING ERROR
           LOAD "Test"                   <tape.dev>_SPT_0_Test
           LOAD "Test2"                  <tape.dev>_SPT_0_Test2
           LOAD "Income" DATA x()        <tape.dev>_SPT_1_Income
           LOAD "DBase" DATA d$()        <tape.dev>_SPT_2_DBase
           LOAD "Pattern"SCREEN$         <tape.dev>_SPT_3_Pattern
           LOAD "Pattern"CODE            <tape.dev>_SPT_3_Pattern
           LOAD "Tables"CODE             <tape.dev>_SPT_3_Tables
           LOAD "Tables"CODE 33000       <tape.dev>_SPT_3_Tables
           LOAD "Tables"CODE 65000,536   <tape.dev>_SPT_3_Tables
           LOAD "Tables"CODE 65000,500   ...TAPE LOADING ERROR
```

{Note: the clever ones amongst you may figure out how get rid of the 'TAPE LOADING ERROR' report at LOAD ""; do you know which file is opened?}

There are several other circumstances in which a 'TAPE LOADING ERROR' can occur. First, if Spectator cannot open the file you have requested, this error will be displayed. Secondly, there are some cases in which Spectator CAN open the file, but - after having read the header info - the ZX Spectrum ROM may decide that the contents are no good (e.g. because a file is too long to be stored in the buffer space you have available).

Spectator will never exceed address 65535 during LOAD. The Spectrum ROM would "wrap" the rest of the file to address 0 onwards, but Spectator simply stops LOADing. This protects the ROM area against an accidental overwrite.

On the normal ZX Spectrum, LOADing and SAVEing to/from tape is case-sensitive, so, in the above examples, LOAD "test" would fail to LOAD as the program was SAVEd with an upper-case 'T'. Spectator does some clever things to make sure that this case-sensitivity remains unnoticed, so you can just as well type LOAD "test" or LOAD "tEsT" etc. All of these commands will properly LOAD the file '<tape.dev>_SPT_0_Test'.

Everything that has been said for LOAD applies equally well for MERGE, except ofcourse that you can only MERGE 'type 0' files (basic lines & variables).

Application programs in machine code which call the Spectrum ROM code to SAVE data to tape will work just fine. However, LOADing data from machine coded application programs will not always work (even when they call the ROM) as Spectator has no possibility to work out the name of the required file. For LOAD, the Spectrum ROM uses a buffer of 2 times 17 bytes; the first part contains the

page 29

required header info, the second part is used to store a header loaded from tape. When an application program mimics this procedure, the LOAD will be successful (provided the file actually exists). In all other cases, a 'Tape Loading Error' is bound to occur.

(2) TAP tapefile support

The TAP-file is yet another way to store tapedata on disk. The TAP file was invented by Gerton Lunter for his 'Z80' Spectrum emulator for PC's and comes in two flavours: the simple TAP file, which is basically the same as the SPT file of the previous section, and the multiple TAP++ file, which is more flexible in many respects. You can use only one tapefile type (either SPT, or TAP, or TAP++) at any moment; use the menu system (F1, option 'T', item 'T') to select the tapefile type you prefer.

In case of the simple TAP type, SAVE and LOAD commands will write (read) headers and data to (from) single files which are named

```
           <tape.dev>_<name>_TAP

with       <tape.dev> : the tape device (FLP1 by default)
           <name>     : the name supplied in the SAVE command
                        (following ZX Spectrum standards, the name
                        is 1..10 characters long)

Example:   SAVE "Test"         creates   <tape.dev>_Test_TAP
           LOAD "Pattern"CODE  accesses  <tape.dev>_Pattern_TAP
```

It is clear that the filetype is not a part of the name of the tapefile, so it is NOT possible to SAVE a basic file and a code file with the same name...

For extensive notes about <tape.dev>, verifying files, LOAD "", tape loading errors, case sensitivity, machine code loaders and merging files: see the section about SPT-files.

The two main difference between the simple TAP file and the SPT file are: (1) the TAP filename does not contain the filetype, so if you save Basic and Code with the same name, they will overwrite eachother; (2) the file contents of both types are slightly different and therefore mutually incompatible. Therefore, I recommend not to use the simple TAP files, unless you wish to port tape files from or to MS/DOS.

The multiple TAP++ filetype is -- though very similar to the simple TAP type -- much more flexible. When the TAP++ type is selected, all tapedata will be read from and written to a single file, which name can be specified by the user. This means that application

programs or games comprising many (small) parts can conveniently be stored in a single file. Also headerless files or files with alien headers can be handled by the TAP++ fileformat. The TAP++ is a good representation of the actual tape used by the Spectrum.

page 30

The file used for storing and retrieving data in TAP++ tape mode can be selected via the menu system (F1, 'C', 'U') [Note that you will ALSO have to select TAP++ tapefile mode via main menu option 'T'.] Each time you change the TAP++ filename (or simply reconfirm it), the filepointer will be set to the beginning of the file (i.e. the tape gets rewinded); the same happens if during loading the end of the file is reached (the TAP++ file acts like a closed loop of tape). Saved data will always be appended to the end of the file rather than be inserted at the current filepointer.

Typing LOAD "" in TAP++ mode will load the first Basic program encountered. The TAP++ mode does not cure the case sensitivity of the Spectrum. If you try to load a file which is not present in the current TAP++ file, the tape will loop around forever, showing the names of the programs it encounters during the search; just press Break to stop this process.

(3) MICRODRIVE support

Spectator was the first Spectrum emulator worldwide (for any system) with microdrive support. The microdrive format of Spectator is now being used in other Spectrum emulators as well, for example 'Z80' (PC) and 'ZM/x' (QL).

The MicroDrive subroutines and matching Hook codes of the ZX Interface 1 Shadow ROM are fully emulated. However, it is not possible to read ZX Spectrum microcartridges directly via the QL microdrives. The files on your Spectrum microcartridges must be transported to the QL in another way, e.g. via the serial line (see next section).

All eight microdrives can be used. Spectator puts the contents of each 'microcartridge' in a single file of about 135 kilobytes (270 QL-sectors). The names of the accessed files are as follows:

```
*"m";1;   accesses  <mdv1.dev>_MicroDrive_1
*"m";2;      "      <mdv1.dev>_MicroDrive_2
*"m";3;      "      <mdv1.dev>_MicroDrive_3
*"m";4;      "      <mdv1.dev>_MicroDrive_4

*"m";5;   accesses  <mdv2.dev>_MicroDrive_1
*"m";6;      "      <mdv2.dev>_MicroDrive_2
*"m";7;      "      <mdv2.dev>_MicroDrive_3
*"m";8;      "      <mdv2.dev>_MicroDrive_4
```

```
with      <mdv1.dev> : 1st microdrive device (RAM1 by default)
          <mdv2.dev> : 2nd microdrive device (RAM2 by default)
```

The microdrive devices can be changed via the menu system. First select option 'C' (change defaults) and then option 'M' or 'N' (1st/2nd microdrive device). You can edit the current setting and accept it by Enter. Pressing ESC (or cursor up/down) will leave the original devicename unchanged. Spectator will not check the

page 31

existence of the device you enter.

As can be seen, Spectator stores up to 4 'microdrives' on a single device; this requires a capacity of 4*270 = 1080 QL-sectors, or 540 kilobytes.

A microdrive-file is generated as soon as a microdrive is accessed from Spectator. Initially, a pre-formatted, empty cartridge (read: file) with the name 'Nomenescio' will be produced ('Nomen Nescio' means: 'name unknown'). The actual data sectors are filled with

rubbish, but the headers and record descriptors are clean and the full capacity of 127 kilobytes is available. If the file cannot be accessed or generated for some reason, a 'microdrive not present' error will be returned by the Spectrum.

I strongly advise to re-format this cartridge yourself using the proper Interface 1 routines, so the most sensible command to begin with is 'FORMAT'. In Spectrum BASIC type:

```
FORMAT "m";1;"<Name>"
```

with <Name> the cartridge name (1 up to 10 characters). This will properly set up the file <mdv1.dev>_MicroDrive_1. The formatting and checking is fully performed from the Interface 1 ROM and (this takes about 6 seconds on a QL with Gold Card and 3 seconds on a QXL). You can ensure yourself that the format has succeeded by typing:

```
CAT 1
```

This should display the cartridge name and the number of kilobytes available for file storage (should be 126!). Obviously, no files are present as yet. Now, microdrive 1 can be used for SAVE, VERIFY, LOAD, MERGE, OPEN, CLOSE, MOVE, ERASE and any other file manipulations.

Note that microdrive filenames are case-sensitive; contrary to the case of SPT-files, this sensitivity is NOT cured by Spectator in any way so as to ensure best compatibility.

The contents of each microdrive file are as follows:

```
           /   15 bytes   header             \
           |   15 bytes   record descriptor  |
254 times  <                                 >  + 1 = 137923 bytes
           |  512 bytes   data block         |
           \    1 byte    data checksum      /
```

The last '+1' byte contains the 'write protection' flag. This byte is not yet being used, it's only appended. Older microdrive files may not have this additonal byte but Spectator can handle them equally well.

The 12-byte preambles of each header and record descriptor

page 32

(entirely useless for our purpose) are stripped off, saving a total of 6096 bytes. It is obvious that these microdrive files always offer 254 free sectors (or 127 kilobytes, the maximum amount that can handled by the ZX Interface 1 ROM). This is quite a lot more than the 160 sectors (80k) guaranteed by Sinclair for the real thing!

There is no objection whatsoever against copying or renaming microdrive files. You can clone Microdrive 1 to Microdrive 2 simply by typing:

```
COPY <mdv1.dev>_MicroDrive_1 TO <mdv1.dev>_MicroDrive_2
```

This makes backing-up quite easy. I also want to remind you that MicroDrive_1 effectively becomes MicroDrive_5 when you move the file from <mdv1.dev> to <mdv2.dev> (and vice versa).

(4) Z80 compressed file support

An excellent ZX Spectrum emulator called 'Z80' has been written by Gerton Lunter for the MS/DOS PC. This emulator is quite popular and a HUGE number of ZX Spectrum programs in MS/DOS format can be obtained from BBS 'Tatort' (Dick Pluim):

'Tatort' BBS, Groningen, Netherlands: +31-50-264840

v22, v22bis, v32, v32bis, MNP2-5, v42, v42bis (300-14400 baud)

The BBS stocks a lot of Spectrum software, Spectrum emulators for QL, PC, Atari and Amiga, and Spectrum-related utilities for PC. At the time of writing the BBS is open 24 hours a day, but this is subject to change. Please try calling between 22:00 and 9:00 local time first.

Z80 files are in fact compressed 'snapshots' of the full 48k or 128k RAM of the Spectrum, including screen, system variables, basic listing, basic variables, machine code, machine stack and any data above RAMtop. A small header contains information about the current values of all Z80 registers (including PC and SP) as well as some emulator settings (e.g. synchronization) so the program can be properly resumed after it has been LOADed.

Spectator supports these special files (which have extension .Z80 in MS/DOS). After having transported a Z80-file called '<name>_Z80' from MS/DOS to Qdos (using, for example, XOVER, ATR device driver, AtariDOS extensions or MultiDISCover) RENAME it to 'Z80_<name>'. Once in the Spectrum emulator, you can LOAD this file by typing:

LOAD "/<name>"

The forward slash ('/') tells Spectator that a Z80 file is wanted. Please note that <name> itself cannot exceed 9 characters, but this should be no problem since the MS/DOS limit is 8 characters.

Example: LOAD "/Booty" loads <snap.dev>_Z80_Booty
page 33

with <snap.dev> : the snapshot device (FLP1 by default)

The snapshot device can be changed via the menu system. First select option 'C' (change defaults) and then option 'Z' (snapshot device). You can edit the current setting and accept it by Enter. Pressing ESC (or cursor up/down) will leave the original devicename unchanged. Spectator will not check the existence of the device you enter.

When the Z80 file cannot be opened, you will get a 'TAPE LOADING ERROR'. Any extension to the LOAD command (e.g. CODE or DATA) will be ignored. When the file is found, Spectator automatically switches to Keyrow mode since most of the Z80-files are games. Press CTRL-F3 or use the menu system for setting Keyboard Queue Mode.

The header of a Z80-file created by Z80 version 1.40 and onwards contains a bit that indicates whether or not an Issue-2 Spectrum is to be emulated. The main difference between Issue-2 and later Spectrums is the keyboard handler and some older games may only work on an Issue-2 Spectrum. Spectator reads the Issue-2-bit and will apply the proper keyboard decoding. Issue-2 decoding remains active until Spectator is reset by way of CTRL-ALT-ESC. Issue-2 emulation may also be changed via the pop-up menu system (F1, option T, key 2).

Spectator can handle both the old-type and new-type Z80 files. The old-type Z80 files could only contain 48k snapshots; the new-type snapshots can also contain 128k snapshots. Spectator will automatically select the proper hardware mode when loading an old- or new-type Z80 snapshot.

Besides reading them, Spectator can also create new-type Z80 files. This implies that you can easily transfer Spectrum programs running in Spectator to the Z80-emulator for the PC or to any other Spectrum emulator supporting the Z80 file format. Please refer to the section 'Pop-Up Menu System' for more details about Z80 snapshot saving.

(5) Magic Box and DISCiPLE memory snapshot support

The "Magic Box" is a Russian disk interface for the Spectrum. Amongst other things, it can write a full 48k memory snapshot to disk. A small part of the 1k header contains information about the current values of most Z80 registers, so a program can be properly resumed after it has been LOADed. Magic Box Files can be loaded by Spectator if they are stored on disk with the prefix 'MBF_'.

The well-known DISCiPLE disk interface has a similar option. It can write SNAP files to Disciple disks. These files can be converted to QL 'SNP' files by way of the utility 'ddConvert' (see section 'Conversion Utilities'). This creates files with the prefix 'SNP_' and these can be loaded directly into Spectator.

page 34

To read such files into Spectator, type:

```
            LOAD "?<name>"      for an  MBF  file
            LOAD ".<name>"      for an  SNP  file

Example:    LOAD "?FireLord"    loads <snap.dev>_MBF_FireLord
            LOAD ".SNAP-C"      loads <snap_dev>_SNP_SNAP-C
```

(see section 'Z80 compressed file support' for information about <snap.dev>). When the MBF- or SNP-file cannot be opened, you will get a 'TAPE LOADING ERROR'. Any extension to the LOAD command (e.g. CODE or DATA) will be ignored. When the file is found, Spectator automatically switches to Keyrow mode.

CONVERSION UTILITIES (!!! registered versions only !!!)

Another way to get Spectrum files on QL disks is to use one of the conversion utilities written by Jack Raats, which are part of a registered Spectator package. These allow you to transfer files from your original Spectrum disks to QL disks in the SPT_ format of Spectator. Also, utilities to convert the P-files of the Dutch HCC/SGG/Impuls association to SPT-files and vice versa are included.

```
P2SPT_exe      : unpacks a P-file, creating a number of SPT-files
               : usage:  EW P2SPT_exe;'dev1_pfile dev2_'
                         dev1_pfile : source device and P-file name
                         dev2_      : destination device

SPT2P_exe      : packs a number of SPT-files in a single P-file
               : usage:  EW P2SPT_exe
                         and answer the questions appropriately

bdConvert_exe  : converts the files on a BETA disk to SPT files
               : usage:  EW bdConvert_exe;'dev1_ dev2_'
                         dev1_ : device containing Beta disk
                         dev2_ : device containing Qdos disk

ddConvert_exe  : converts the files on a DISCIPLE disk to SPT files;
                 also converts SNAP files, which can be loaded into
                 Spectator immediately (see section 'File Management').
               : usage:  EW ddConvert_exe;'dev1_ dev2_ step'
                         dev1_ : device containing Disciple disk
                         dev2_ : device containing Qdos disk
                         step  : number of tracks to step
                                 1 for 40-track disk in 40-track drive
                                 2 for 40-track disk in 80-track drive
                                 1 for 80-track disk in 80-track drive

odConvert_exe  : converts the files on an OPUS disk to SPT files
               : usage:  EW odConvert_exe;'dev1_ dev2_'
                         dev1_ : device containing Opus disk
                         dev2_ : device containing Qdos disk
```

page 35

RS232 SERIAL PORT

The ZX Interface 1 contains a serial port that can be accessed by way of the "b" and "t" channels. Spectator supports these channels, simply by redirecting all Spectrum RS232 output to the QL serial port (SER1hr, so the handshake lines must be wired!) and redirecting all the input on this same port to Spectrum Basic. It is not (yet) possible to use other QL devices (e.g. SER2, a parallel port or even a file) for this purpose.

The baud rate of the serial port is set according to the delay-period held in the two-byte system variable BAUD (address $5CC3). The baud rate of the QL is fixed to standard values, so illegate baud rates will be rounded to the nearest legal one. The available baud rates are 300, 600, 1200, 2400, 4800, 9600 and 19200. The rates 50 and 110 are not possible on the QL and are replaced by 75 baud. Also, take care with 19200 baud, as the QL can only transmit at this rate (i.e. not receive)!

It is NEVER sufficient to set the required baud rate OUTSIDE Spectator before printing or communicating; it is ALWAYS necessary to use a FORMAT "b" command within Spectator to set the proper baud rate used by Spectator. By default, the Spectrum (hence also Spectator) uses 9600 baud.

You can use the "t" channel to print texts or to list your programs to a printer connected to SER1hr. As ZX Interface 1 edition 2 is included, you can set the page width of your paper by POKEing address 23729; default value is 80 columns.

You can use the "b" channel to send and receive binary data to and from SER1hr. If you have a Spectrum with Interface 1, you can connect it to the QL and, after having set both to equal baud rate (preferably 9600 baud, the default value), transmit programs from the Spectrum directly into Spectator (and subsequently store them on disk), e.g. by typing

```
SAVE *"b"           on the Spectrum
LOAD *"b"           on Spectator
```

and similarly for array- or code-files. When the working speed is set to 9600 baud, the effective transfer speed will be about 550 bytes per second (measured on a QL with GC). It is ofcourse also possible to transfer data from Spectator to the Spectrum, e.g. by typing

```
SAVE *"b"           on Spectator
LOAD *"b"           on the Spectrum
```

In this case, the effective transfer speed is as low as 1600 baud. So, this is not the fastest file tranfer you can think of, but it does work and it's interesting in the sense that a real Spectrum actually communicates with an emulated one!

page 36

While Spectator is sending or receiving data (or waiting for data to be sent or received), the Spectrum screen may be frozen (the logo will stop). Don't worry: this is supposed to happen! RS232 activity can always be interrupted by pressing Shift-Space or CTRL-ESC. This however may leave the QL RS232 buffer partially filled with data, so subsequent RS232 commands may return error reports like 'wrong file type'.

There is an alternative method for getting data from the Spectrum into Spectator, namely directly via SuperBASIC. For example, a SCREEN$ can be sent from the Spectrum to the QL in the following way:

```
SAVE *"b" SCREEN$             on the Spectrum
COPY ser1hr TO flp1_name      or similarly on the QL
```

When the Spectrum has finished sending the data, press CTRL-Space on

the QL to terminate the COPY command. The file 'flp1_name' then contains the essential data but has an RS232-header instead of the usual tape-header of SPT files. The program 'b2spt_bas' should now be used to convert the RS232 file ('b' file) to an SPT file. This SPT file can finally be loaded into Spectator in the usual way. Although this procedure seems more cumbersome, it is flexible when a lot of data needs to be transferred. Finally note that 'b2spt_bas' can also be used to transfer data received on the SER1hr port directly to an SPT file (without the intermediate 'b' file).

JOYSTICKS

On the Spectrum, three joystick interfaces are being widely used:

(1) Cursor, Protek, AGF (three different names for the same type): the movement switches on these interfaces are simply connected to the usual keys for cursor movement ('5' through '8', unshifted though). The fire button is connected to key '0'.

(2a) Sinclair ZX Interface 2, #1 (lefthand side): the movement switches are connected to the keys '1' through '4', the fire button is connected to key '5'.

(2b) Sinclair ZX Interface 2, #2 (righthand side): the movement switches are connected to the keys '6' through '9', the fire button is connected to key '0'.

(3) Kempston: the only interface that is not connected to the keyboard. The switch settings must be read by way of IN instructions.

Most games support at least one of these four interfaces. Spectator supports ALL these joystick types. If you select any one of the above joystick types (via menu option 'J', see section about the pop-up menu system), the cursor keys (up, down, left, right) and space bar (fire) are reprogrammed in such a way that they respond as the joystick you have chosen. This means that a QL-joystick connected to CTL1 can then be used as a Spectrum joystick!

page 37

Obviously, the joystick emulation ONLY works if BOTH the game AND Spectator are set to the SAME type of joystick.

Note that the selected joystick is only active in keyrow-mode ('#'). Below you can see how the five relevant keys of the keyboard respond in keyrow-mode when any of the joystick selections is active:

None = keyboard mode	Left	Down	Up	Right	Space
Cursor, Protek, AGF	5	6	7	8	0
Kempston	-	-	-	-	-
Sinclair ZX-IF-2, #1	1	3	4	2	5
Sinclair ZX-IF-2, #2	6	8	9	7	0

For the case of Kempston, the five keys are fully disabled: they can only be 'seen' via the proper IN-port.

If a game supports the ZX-Interface-2, this is usually denoted by names such as 'Sinclair', 'Sinclair 2', 'Interface II' etc. In most cases it is not specifically mentioned whether the lefthand or the righthand port of the interface is to be used; this usually means that the righthand side (#2) needs to be used. In some cases, both sides can be used 'at the same time'. Alternatively, some games cleverly ask you to press the fire-button, so that it can see itself to which side your joystick is connected or even what kind of joystick you are using!

STOPPING

There are three legitimate ways to end your Spectator session. Directly

from Spectrum BASIC, you may type the following command:

```
LOAD " STOP "
```

with STOP typed as a token (SymbolShift-A on Spectrum, so CTRL-A on QL). Alternatively, you can pop up the main menu (press F1) and use the Quit option (press Q). Note that an additional confirmation is requested.

If you wish to stop Spectator from OUTSIDE the program itself, this can be accomplished by removing the job 'Spc_Vita'. In SuperBASIC this can be done by typing

```
RJOB SPC_VITA
```

You can also use the RJOB option of QPAC-2 (or any equivalent program). Take care NOT to delete any other Spectator job as this will crash your system! Next, switch to the Spectator screen by CTRL-C. This will clean up the QL, removing all channels, jobs and interrupts owned by Spectator. If Spectator was set to Keyboard-Queue Mode, CTRL-C is NOT needed as Spectator will be fully removed directly after the RJOB.

Z80 EMULATION

page 38

As far as I have been able to test, all Z80 instructions are emulated correctly: since the ZX Spectrum ROMs are the basis for Spectator to function, the latter statement should be true. Moreover, all tested (machine code) programs (both games and utils) are working just fine. In an official review of Spectrum emulators for the QL by Simon N. Goodwin, Spectator was the ONLY one receiving an 'A' for compatibility: a flattering confirmation of Spectator's thoroughly debugged Z80 emulation!

Some brief remarks for Z80 freaks:

+ The Half Carry and Subtract flag of the Z80 flag-register (f) are NOT emulated. Even though nearly all Z80 instructions alter these flags, there is only one instruction that uses the result, viz. DAA. Since DAA is used only once in the ZX Spectrum ROM and since proper emulation of the Half Carry greatly reduces operation speed of Spectator, I think this exclusion is a sensible decision.

+ Despite the previous remarks regarding the half-carry flag, DAA is still emulated fairly reliably. In most cases, DAA will be preceeded by an instruction which adds data to or subtracts data from the accumulator (A), the decimal adjust taking place immediately after. Instead of working out the half-carry flag setting caused by these instructions, the 'class' of the instruction and (when applicable) the original value of the accumulator, the argument and the original setting of the carry-flag are stored in such a way that a subsequent DAA can always reconstuct the initial conditions and perform the proper MC68000 'ABCD' or 'SBCD' directly (instead of using the cumbersome half-carry method). Currently, 7 instruction classes are supported: 'ADD A', 'ADC A', 'INC A', 'SUB', 'SBC', 'DEC A' and 'NEG'. Any other instruction preceeding the DAA will cause unpredictable results. In the Spectrum ROM, the DAA occurs after an 'ADC A,A' and is therefore emulated properly.

+ Many Spectrum games use unofficial Z80 instructions; the unofficial instructions that are emulated by Spectator are SLL and BIT 7 as well as instructions acting on the low or high bytes of IX and IY.

+ It seems that many Spectrum games try to change the ROM area for some reason or another (maybe piracy protection). These ROM pokes are ignored by Spectator. The ROM protection covers virtually all memory operations. For reasons of speed, an exception has been made for stack manipulations: these are not 'protected', so you could change the ROM by setting the SP in the ROM area and PUSHing registerpairs.

I assume/hope that no game will actually do this... By the way, the Spectrum ROM also pokes into its own code! This is obviously protected as well.

+ Some programs make use of floating IN-ports for timing purposes; such ports are not connected to any hardware device and will return either screen data or just the value $FF. Spectator mimics the temporal behaviour of these ports in a highly accurate degree and returns the right values at the right time. Try Arkanoid: no other emulator I've
page 39
seen gives such a stable 'bat'!

GENERAL COMPATIBILITY -- some remarks

Since the original ZX Spectrum ('128) ROMs and ZX Interface 1 Shadow ROM (edition 2) in Z80 code are included in the Spectator executable, almost complete software compatibility is guaranteed. Almost, since only some VERY specific software can obviously NOT be emulated. Application programs which properly use the Hook codes of ZX Interface 1 will function just fine. Programs that use their own INs and OUTs for microdrive access, serial communication, tape reading or any other non-standard I/O activity will certainly fail to operate. Apart from these exceptional applications, ANY program running on a real ZX Spectrum ('128) should ALSO run on Spectator. Note again that also Spectum '128 addressing modes (with all their pecularities) are emulated by software with the highest possible accuracy and cannot possibly lead to program failure.

SPECTATOR vs SPECTRUM

Spectator has some advantages and disadvantages with respect to the real ZX Spectrum. To name just a few of the many advantages: safe housing in a well stabilized computer, improved keyboard and screen quality, full 48k or 128k RAM available, reliable and fast data storage (both tape, microdrive and snapshots), very easy hardware selection (48k, 128k, ZXIF1, joystick interface), etc., etc.. Additionally, Spectator happily MultiTasks, so (if you wish) you can even run several copies of Spectator at one time!

If you have a standard QL with a Trump Card, a main disadvantage is SPEED! A common property of all emulators known to man is the general lack of speed. Since single Z80 instructions are being translated in at least five and sometimes as much as tens of MC68000 instuctions, Spectator is bound to be slow. The Spectrum emulator cannot do better than about 8% of the original ZX Spectrum speed. I therefore recommend NOT to run Spectator on a 'normal' QL. You will only get irritated by the low speed!

Much better performance is obtained on a QL with a Gold Card: the speed increases to about 32% (one third of original Spectrum speed). Although 32% may still seem terribly slow, this speed is sufficient for general editting purposes and for running most (machine code) games and utilities. In fact, the ZX Spectrum is quite a fast computer and many machine-code games use fixed delay-loops or HALTs for proper human operation. Taking out the delay-loops will make most games as fast and exciting as they were on the ZX Spectrum.

Top performance is obtained on a QL with a Super Gold Card and on a QXL (both with cache on). The average emulation speed is about 130%, i.e. significantly more than the Spectrum itself. It's a real pleasure to use Spectator on these machines: it runs very fast without loss of its excellent (and renowned) compatibility.
page 40

OTHER SPECTRUM EMULATORS

The following list of Spectrum emulators was taken from the manuals of

JPP and Z80 (see below) and appended where possible. The emulators with unknown name and/or author are not included in the list. I welcome any information about existing Spectrum emulators not mentioned below.

name	system	author	remarks
JPP	PC	Arnt Gulbrandsen	needs 386+ and VGA+, weird screen
SIMULATOR	C64	Whitby Software	emulates Spectrum BASIC only
SP	PC	Swiatek & Makowski	crashprone, no documentation
SPECEM	PC	Kevin J. Phair	needs 16 MHz 286+ and EGA+
SPECTATOR	QL	Carlo Delhez	you're reading its manual now!
SPECTRUM	Amiga	Peter McGavin	needs 25 MHz Amiga for full speed
SPECTRUM	PC	Pedro Gimeno	has tape interface & pokeable ROM
SPECULATOR	QL	various authors	PD, quite fast, few features
VGASPEC	PC	Alfonso Olloqui	needs 286+ and VGA+, no manual
Z80	PC	Gerton Lunter	handles any processor/video card
ZM/1..3,HT	QL	Ergon Development	ZM/HT is a fast Spectrum compiler
ZX	QL	Andrew Lavrov	made in Russia

As quite a lot of Spectator users have asked me where they can get the excellent Spectrum emulator 'Z80' for the PC, here's the address:

Gerton Lunter,
P.O. Box 2535,
NL-9704 CM Groningen,
Netherlands.

I have included this address here for your benefit. So, you can do me a favour in return: when you write to him, please mention the QL and Spectator somehow (and don't forget to send him at least one IRC for answering you)!

WARRANTY

During the development of Spectator, lots of existing Spectrum programs have been loaded into Spectator as to test their behaviour in the emulator environment. All these programs were working just fine.

However, if you should come across a program that does not work properly with the current version of Spectator, I'd be most glad to hear from you! If Spectator is to blame, you'll get a repaired update for free (provided I can find the 'bug'). This way you are helping me locating bugs and helping all other Spectator users in getting a better emulator! Please send me a disk with the program(s) (plus brief instructions) on it (the disk will ofcourse be sent back to you).

FINAL NOTES

page 41

 QXL and (Super) Gold Card are available from Miracle Systems Ltd, York, UK. Thanks to Jack Raats for providing me with useful hardware, paperware, software, conversionware and brainware. Thanks to Gerton Lunter for interesting correspondence regarding Spectrum 128k and many other topics, and for his kind support of Spectator. For more information on Spectator, do not hesitate to contact me at:

Carlo Delhez,
Emmastraat 3,
4651 BV Steenbergen,
Netherlands.

Thanx 4 reading the manual & 4 using Spectator !!

Alas... End of file

SPECTATOR TECHNICAL SPECIFICATIONS   version 0.14, October 1993

Compilation by Carlo Delhez, copyright 1991-1993

All amendments or comments to this document are welcome!
Updated 25/3/93 with SPECULATOR details by Simon N Goodwin.

Carlo Delhez, Emmastraat 3, 4651 BV Steenbergen, Netherlands.
E-Mail (Internet): tnndcarlo@cycl.phys.tue.nl (until May 1994).

Contents:

Note: entries marked with ¿ these signs ? are uncertain or unknown!

"B" FILE FORMAT (RS232 header file)   (source: private investigations)

Files that are being sent over the 'B' (RS232 binary) channel of the Spectrum Interface 1 have a 9-byte header (different from the 17-byte tape header). The program 'B2SPT_bas' converts a file with RS232 header to a file with a tape header; the 'source file' may also be taken from RS232 directly!

offset	length	contents
$00	byte	filetype: value $00 = basic plus variables, $01 = numerical array, $02 = string array, $03 = code block
$01	word	length of datablock (LSB first)
$03	word	original storage address of datablock (LSB first)
$05	word	filetype $00: total length of BASIC program (LSB first) $01: not used $02: not used $03: not used
$07	word	filetype $00: linenumber for autorun (LSB first, value $FFFF if no autorun) $01: not used $02: not used $03: not used
$09		start of datablock (length determined by offset $01)

MBF MAGIC BOX FILE FORMAT (source: private communications with Andrew Lavrov)

The Magic Files are generated by a Russian diskinterface for the Spectrum called 'the Magic Box'. A magic file contains a 48k snapshot of the Spectrum RAM memory, preceeded by a 1k header containing the values of all Z80 registers at the moment the snapshot was made.

offset	length	contents
$000	word	BC register (order: C,B)
$002	word	DE register (order: E,D)
$004	word	HL register (order: L,H)
$006	word	BC' register (order: C',B')
$008	word	DE' register (order: E',D')
$00A	word	HL' register (order: L',H')
$00C	word	SP, Stack Pointer (LSB first)
$00E	word	IX register (LSB first)
$010	word	IY register (LSB first)
$012	byte	I register (if value is $3F then IM 1, else IM 2)

page 43

offset	length	contents
$013	byte	¿ not used ? (value $00)
$014	byte	IFF1
$015	byte	R register
$016	byte	IFF2
$017	word	¿ first address of snapshot ? (value $4000, LSB first)
$019	word	¿ last address +1 of snapshot ? (value $0000, LSB first)
$01B	997 bytes	not used (contents $E5)
$400	48k	memory snapshot

Once the snapshot is loaded into the Spectrum memory, the remaining registers can be found on the machine stack:

(SP+0)	byte	F' register
(SP+1)	byte	A' register
(SP+2)	byte	F register
(SP+3)	byte	A register
(SP+4)	word	PC, Program Counter

When these values are fetched, the SP must be increased by 6. This happens automatically when POPs are used to clear the stack.

MICRODRIVE FILE FORMAT (source: 'Spectator' manual, and the Spectrum Shadow ROM disassembly by Gianluca Carri)

A Microdrive File contains 254 'sectors' of 543 bytes each, so the total length is 137922 bytes. The contents of each 'sector' are as follows:

offset	length	contents
$000		15-byte header
$000	byte	header identifier (value $01)
$001	byte	sector number (values $FE down to $01)
$002	word	not used
$004	10 bytes	microdrive cartridge name (space filled)
$00E	byte	header checksum (of bytes $000 through $00D)
$00F		15-byte record descriptor
$00F	byte	filetype (value $00 if sector is not in use)
		- bit 0 : always 0 to distinguish from header
		- bit 1 : set for the EOF block
		- bit 2 : reset for a headerless (data-) file
		- bits 3-7 : ¿ not used ? (value 0)
$010	byte	data block sequence number (value starts at $00)
$011	word	data block length (value max. 512, LSB first)
$013	10 bytes	filename
$01D	byte	record descriptor checksum (of bytes $00F through $01C)
$01E	.5k	data block

$21E	byte	data block checksum (of bytes $01E through $21D, even when data block length is less than 512)

The datablock with sequence number $00 of a header-file starts with a 9-byte

page 44

header, similar to the 9-byte RS232 header (see section on "B" file format):

$00	byte	headerfiletype: value $00 = basic plus variables, $01 = numerical array, $02 = string array, $03 = code block
$01	word	total length of actual data (LSB first)
$03	word	for types $00 and $03: start address (LSB first)
$05	long	filetype-dependent information
$09		start of actual datablock

The three checksums are to be calculated in the following way:

- start with checksum zero
- add databyte to checksum byte
- add (carry+1) to the checksum byte
- if the checksum byte became zero by the last addition then leave it zero, otherwise subtract 1 from checksum byte
- repeat for all data bytes

It may easily be verified that this procedure is a 'MOD 255 addition', hence will never end up with a checksum equal to $FF.

SNP DISCiPLE SNAPSHOT FILES (source: information from Jack Raats)

The Disciple diskinterface can generate two types of snapshots: a 48k snapshot and a 128k snapshot. The actual snapshot is preceeded by a 43-byte (or 44-byte) header containing the values of most Z80 registers at the time the snapshot was made.

offset	length	contents
$00	byte	file descriptor - value = $05 : 48k snapshot - value = $09 : 128k snapshot
$01	10 bytes	filename
$0B	10 bytes	HD00 system variables; zero-filled for snapshots
$15		start of snapshot register area
$15	word	IY register (LSB first)
$17	word	IX register (LSB first)
$19	word	DE' register (order: E',D')
$1B	word	BC' register (order: C',B')
$1D	word	HL' register (order: L',H')
$1F	byte	F-register (not used, bug: was supposed to be F')
$20	byte	I-register (not used, bug: was supposed to be A')
$21	word	DE register (order: E,D)
$23	word	BC register (order: C,B)
$25	word	HL register (order: L,H)
$27	byte	F-register (but not used)
$28	byte	I-register; if I=0 or I=$3F assume IM 1, else IM 2
$29	word	SP, stack pointer, LSB first

- for 48k snapshots (see offset $00):

$2B	48k	memory snapshot

page 45

- for 128k snapshots (see offset $00):

$2B	byte	bankswitch flag byte
$2C	128k	snapshots of the banks 0 thru 7 (16k each)

Once the snapshot is loaded into the Spectrum memory, the remaining registers can be found on the machine stack:

(SP+0)	byte	F register (only bit 2 used for IFF1=IFF2 setting)

(SP+1)	byte	R register
(SP+2)	byte	F register (this is the valid one!)
(SP+3)	byte	A register
(SP+4)	word	PC, Program Counter, LSB first

When these values are fetched, the SP must be increased by 6. This happens automatically when POPs are used to clear the stack.

A 'SNP' file can be generated by the program 'ddConvert' as supplied with registered versions of Spectator. The length of a 48k SNP file should be 49195 bytes: 43 bytes header plus 48k data. The length of a 128k SNP file (not supported by Spectator) should be 131116 bytes: 44 bytes header plus 128k data.

The .PRG SnapShots extracted by the PC program PDEX (part of the SPECEM Spectrum emulator for PC by Kevin Phair) are not directly compatible to SNP files extracted by 'ddConvert'. The .PRG files have a 256-byte header of which only 43 bytes need actually be used. A .PRG file can easily be converted in to a SNP file by removing the non-used data blocks:

$0000	11 bytes	file descriptor and filename
$000B	199 bytes	not used, to be removed
$00D2	32 bytes	HD00 system variables and snapshot register area
$00F2	14 bytes	not used, to be removed
$0100	48k	memory snapshot

Note that .PRG-files may be either 49408 or 49726 bytes long (depending on the treatment of the disksector containing the last 192 bytes of the snapshot). The 318 extra bytes of the longer version may be ignored.

SPECULATOR file formats (source: Speculator documentation)

Speculator supports standard ZX file types 0-3 by mapping them onto unused Qdos file-types 3-6, adding ZX-specific information in the Qdos file header:

ZX-type	Qdos-type	contents
0	3	ZX BASIC program (may include variables)
1	4	ZX BASIC numeric array
2	5	ZX BASIC character array
3	6	ZX BASIC CODE or SCREEN$ file
5	11	GDOS 48K SNAP

page 46

The length of the ZX file can be inferred from the length in the QL header, but -64, the number of bytes used for the header itself.

The normally unused 'extra' field at byte offset 10 in the Qdos file header holds the remaining information from the ZX file header:

Bytes 10 and 11 hold the start address for CODE or the start line for BASIC, in Z80 order, LSB first. If the start line exceeds 9999 no auto-start is performed.

The name of a saved array is encoded at byte offset 11 of the file header. To decode this, AND the value with 31 and add 64 to convert to ASCII capitals.

File type 11 is a Qdos equivalent of the Spectrum G(+)DOS SNAP 48K file-type, with register information added at the end of the 48K Spectrum memory image:

offset	length	contents
$0000	48k	ZX memory image
$C000		start of Z80 register area
$C000	word	IY register (LSB first)
$C002	word	IX register (LSB first)
$C004	word	DE' register (order: E',D')
$C006	word	BC' register (order: C',B')

$C008	word	HL' register (order: L',H')
$C00A	byte	F' register (not always set by GDOS)
$C00B	byte	A' register
$C00C	word	DE register (order: E,D)
$C00E	word	BC register (order: C,B)
$C010	word	HL register (order: L,H)
$C012	byte	F register; P/V flag (bit 2) signals IFF2 (DI/EI)
$C013	byte	I register; IM 2 if I<>63, otherwise IM 1
$C014	word	SP, stack pointer, LSB first
$C016		end of file

Once the snapshot is loaded into the Spectrum memory, the remaining registers can be found on the machine stack:

(SP+0)	byte	F register; P/V flag (bit 2) signals IFF2 (DI/EI)
(SP+1)	byte	R register
(SP+2)	byte	F register (this is the valid one!)
(SP+3)	byte	A register
(SP+4)	word	PC, Program Counter, LSB first

When these values are fetched, the SP must be increased by 6 before re-starting Z80 code emulation.

SPT HEADER FILE FORMAT (source: Spectator manual)

An SPT header file is a junction of the regular Spectrum header and the matching data block, normally written to tape separately. The name of the SPT file already contains the appropriate filetype and the tapefilename.

page 47

offset	length	contents
$00	byte	filetype: value $00 = basic plus variables, $01 = numerical array, $02 = string array, $03 = code block
$01	10 bytes	filename
$0B	word	length of datablock (LSB first)
$0D	word	filetype $00: linenumber for autorun (LSB first, value $8000 if no autorun) $01: not used $02: not used $03: address for loading code (LSB first)
$0F	word	filetype $00: total length of BASIC program (LSB first) $01: not used $02: not used $03: not used
$11		start of datablock (length determined by offset $0B)

TAP TAPE FILE FORMAT (source: manual of Z80 v2.01a and private investigations)

A TAP file is very similar to an SPT file, but slightly more versatile. The TAP file comes in two flavours: the singular TAP file and the multiple TAP file. The former always contains one header block and one data block, the latter can contain an arbitrary mixture of header and data blocks (similar to Ergon ZTA files). Any number of singular or multiple TAP files may be glued together as to compose a new multiple TAP file (unlike Ergon ZTA files). The format below describes a single block in a TAP file; offset is relative to the start of the block rather than relative to the start of the file.

offset	length	contents
$0000	word	length of block (excluding this word); LSB first.
$0002	byte	block descriptor: $00 for header, $FF for datablock; other values for the block descriptor are also possible.
$0003	indef.	block contents

```
$xxxx     byte      block checksum (of bytes $0002 through $xxxx-1)
```

$xxxx is the last byte of the block as determined by offset $0000. The actual block contents of headerblocks are the same as offset $00 through $10 of SPT files. The block checksum byte is the result of a 'cumulative' EOR of all the bytes concerned.

Z80 FILE FORMAT, OLD TYPE   (source: manual of 'Z80' v1.40, by Gerton Lunter)

The old-type Z80 file contains a 48k snapshot of the Spectrum RAM memory, preceeded by a 30-byte header containing the values of all Z80 registers at the moment the snapshot was made plus some additional emulator settings.

```
offset    length    contents

                                 page 48
$00       byte      A register
$01       byte      F register
$02       word      BC register (order: C,B)
$04       word      HL register (order: L,H)
$06       word      PC, Program Counter (LSB first)
$08       word      SP, Stack Pointer (LSB first)
$0A       byte      I register
$0B       byte      R register, bit 7 has no meaning
$0C       byte      - bit 0      : bit 7 of R register
                      bits 1,2,3 : border colour
                      bit 4      : 1 = SamRam switched on
                      bit 5      : 1 = compressed snapshot
                      bits 6,7   : not used
                    - if value is $00 or $FF: replace by $01
$0D       word      DE register (order: E,D)
$0F       word      BC' register (order: C',B')
$11       word      DE' register (order: E',D')
$13       word      HL' register (order: L',H')
$15       byte      A' register
$16       byte      F' register
$17       word      IY register (LSB first)
$19       word      IX register (LSB first)
$1B       byte      IFF1 (0 = DI, else EI)
$1C       byte      IFF2
$1D       byte      bits 0,1 : IM (interrupt mode); value 0,1 or 2
                    bit 2    : 1 = emulate Issue-2 Spectrum
                    bit 3    : 1 = use double interrupt frequency
                    bits 4,5 : video synchronization
                                - value = 0,2 : normal
                                - value = 1   : high
                                - value = 3   : low
                    bits 6,7 : type of joystick
                                - value = 0 : Cursor
                                - value = 1 : Kempston
                                - value = 2 : Sinclair 1
                                - value = 3 : Sinclair 2
$1E                 start of 48k memory snapshot
```

- if the snapshot is not compressed (see offset $0C, bit 5):
    - locations $001E-$C01D contain the 48k memory snapshot

- if the snapshot is compressed (see offset $0C, bit 5):
    - the total file length is indefinite, but the snapshot must be exactly 48k in length after expansion (see below).
    - the last four bytes of the file are $00.$ED.$ED.$00; these bytes are not a part of the memory snapshot and must be ignored.
    - any occurence of the sequence $ED.$ED.$nn.$vv in the file must be replaced by $nn times the byte $vv; note: $nn=$00 may not be used!

Z80 FILE FORMAT, NEW TYPE   (source: manual of 'Z80' v2.0, by Gerton Lunter)

page 49

The new-type Z80 file is an extension of the old-type Z80 file (see above) and can contain either a 48k or a 128k snapshot. The header has been extended and the memory snapshot itself is stored in a slightly different way.

```
offset    length    contents

$00       30 bytes  same as the old-type Z80 file header, but:
                     - offset $06: contains $0000 (rather than the PC)
                     - offset $0C: bits 4 and 5 are no longer used
$1E                 start of additional header
$1E       word      length of additional header, excluding this word;
                    contains +23 in the present case (LSB first)
$20       word      PC, Program Counter (LSB first)
$22       byte      hardware mode
                     - value = 0 : 48k Spectrum
                     - value = 1 : 48k Spectrum plus ZX-IF-1
                     - value = 2 : SamRam (not supported by Spectator)
                     - value = 3 : 128k Spectrum
                     - value = 4 : 128k Spectrum plus ZX-IF-1
$23       byte      in 128k modes, contains last OUT value to port $7FFD
$24       byte      value $FF if ZX-IF-1 is paged in
$25       byte      bit 0 : set if R emulation active
                    bit 1 : set if LDIR/LDDIR emulation active
$26       byte      in 128k modes, contains last OUT value to port $FFFD
$27       16 bytes  contents of the 16 soundchip registers

$37                 start of compressed 16k-snapshot blocks
```

The number of snapshot blocks depends on the hardware mode. In 48k modes, there are three 16k blocks; in 128k modes, there are eight 16k blocks. Each block is a preceeded by a 3-byte block-header (offset is now relative to the start of the block-header):

```
$00       word      length of compressed 16k-snapshot block, excluding this
                    3-byte block-header (LSB first)
$02       byte      block number
$03                 snapshot block, length determined by offset $00
```

The snapshots blocks are compressed in the same way as described for the old-type Z80 file. However, the 4-byte sequence $00.$ED.$ED.$00 is not appended at the end of the blocks. After expansion, each block must be 16k long. There is no end-marker after the last block.

The blocks must be stored in the Spectrum memory according to the block number. The convention is as follows:

```
In 48k modes:       block  8  in area   $4000-$7FFF
                    block  4  in area   $8000-$BFFF
                    block  5  in area   $C000-$FFFF

                    Block numbers other than 4,5,8 should not occur.

In 128k modes:      block  3  in        page 0
                    block  4  in        page 1     note: the position of these
                                     page 50
                    block  5  in        page 2           pages in Spectrum
                    block  6  in        page 3           memory is determined
                    block  7  in        page 4           by offset $23 in the
                    block  8  in        page 5           additional header.
                    block  9  in        page 6
                    block 10  in        page 7

                    Block numbers other than 3..10 should not occur.
```

ZXemu/ZM TAPE FILE FORMAT (source: private investigations)

The tape files used by ZXemu and ZM can contain any number of regular Spectrum headers and/or datablocks, in arbitrary order. The contents of such a file are as follows:

offset	length	contents
$0000	long	tape pointer; points to the start of the header or datablock that is to be accessed by the next tape command; after a rewind, the value is $00000004

Each header and datablock is stored in the following way ('offset' is now relative to the start of this header or datablock within the tape file; the first header file or datablock starts at fileposition $0004):

$0000	long	absolute position within the tape file of the next header or data block
$0004	byte	block descriptor - value = $00 : header - value = $FF : datablock
$0005	indef.	header or datablock; may be of 0 length; length is actually determined by offset $0000

This construction allows ZXemu/ZM to cope with headerless files or files with alien header formats unequal 17 bytes.

REPARACION Y MANTENIMIENTO PREVENTIVO.
APUNTES DE INTERES
======================================

Regla de la pata 1.- Un cable tipo listón consiste en varios conductores tendidos paralelamente formando un cable plano, de donde deriva el nombre de "listón". Uno de los cables de la orilla viene de un color diferente de los demás. Los cables listón son generalmente azules, blancos o grises y el color del de la orilla, más oscuro, como azul oscuro o rojo. Este conductor va al perno numero 1 del conector. ¿Cómo encontrar el perno número 1 en una tarjeta de circuítos¿. Muchos están impresos directamente en la tarjeta. Otros únicamente rotulan el perno 2, dado que el perno 1 está en la parte de atrás de la tarjeta. En algunas tarjetas todos los asientos de cobre para la sodadura en dónde los chips van fijos a la tarjeta de circuíto son redondos, excepto el del perno 1 que es cuadrado. Al igual que en casi todas las reglas existen excepciones. Habrá una pata 1 en una tarjeta de ciruito de disco duro ó diskette, en controladores de disco y en cualquier parte donde un cable tipo listón se conecte a una tarjeta de circuíto.

Calor y shock térmico.- El shock térmico se origina al sujetar los componentes a cambios rápidos de temperatura. Puede hacer fallar la computadora por causa de daños de expansión/contracción. Por ejemplo dentro de una PC la pemperatura puede ser de 55 grados fahrenheit. Al encender la PC se calienta a 120 grados en 30 minutos. Este incremento de 65 grados de temperatra en sólo 30 minutos, produce shock térmico.

Ciclos de trabajo.- Los dispositivos deben disipar el calor tan rápidamente como se genera. Sin embrargo, no todos los dispositivos son capaces de éllo. Se dice que cada dispositivo tiene un ciclo de trabajo, que es un número expresado en forma de porcentaje y constituye la proporción del tiempo que puede trabajar sin quemarse. Los motores de

disco duro pueden trabajar contínuamente sin quemarse ya que tienen un ciclo de trabajo del 100 por ciento. Los motores de las cabezas de los diskettes, en cambio, no están diseñados para operación contínua.

Rango de temperatura.- Los componente electrónicos tie nen un rango de temperatura ambiente dentro del cual deben trabajar. IBM sugiere que las PC, por ejemplo, están construídas para trabajar en el rango de 60 a 85 grados Fahrenheit. Esto se debe a que las tarjetas pueden funcionar hasta los 125 grados, una máquina típica puede calentarse hasta 40 grados sobre la temperatura exterior. 125-40=85 grados , que es la máxima temperatura ambiente recomendada. Además el calor acelera la corrosión. No se aconseja que la temperatura interior de la PC supere los 110 grados.

Polvo.- El polvo está en todos los lados. Consiste en pequeñas partìculas de arena, esqueletos fósiles de criaturas minúsculas que vivieron hace millones de años, piel muerta, partículas de papel, y minúsculos crustáceos llamados ácaros del polvo que se alimenta de los otros componentes. El polvo

página 1

es responsable de varios males. En primer lugar, el polvo se pega a las tarjetas de circuíto de la computadora. Conforme se acumula, toda la tarjeta se cubre de una fina capa aislante de calor, y este aislamiento térmico es malo para las computadoras.

Rayos solares.- La luz directa del sol es mala para el equipo electrónico. La luz solar es también mala para los diskettes. Encuentre un área de sombra ó use cortinas.

Alzas de voltaje autoinducidas.- Son elevaciones de voltaje producidas por el usurario. Cada vez que se enciende un aparato eléctrico se genera un alza transitoria en él. Algunos de los mayores esfuerzos que reciben los aparatos eléctricos consiste en encenderlos ó apagarlos. Los focos se funden generalemnte al apagarlos ó al encenderlos. Un estudio comprobó que cuando se encciende un aparato, consume de cuatro a séis veces su potencia normal durante menos de un segundo.

Transitorios.- Son cambios breves de energía. Puede ser alza ó bajada de voltaje. Las caídas momentáneas y las alzas momentáneas son transitorios. Siendo breve, el transitorio puede ser de una frecuencia lo suficientemente alta que se cuele por los capacitadores y demás cosas de protección que se tengan en la fuente de poder, y llegue a barrenar en los chips. (No son hoyos que se puedan ver, por lo menos sin equipo especial). Los transitorios tiene un efecto acumulativo.

Exceso de voltaje.- Se dice que está presente un exceso de voltaje cuando se tiene más del voltaje normal durante un período mayor a 2.5 segundos. La medida que lo determina toma como promedio fluctuaciones en varios segundos. El exceso de voltaje crónico es tan perjudicial al sistema como un alza de voltaje transitorio. Los chips pueden fallar como resultado de élla.

Bajo voltaje.- Las reducciones de voltaje son malas y confunden a las fuentes de poder. Una fuente de poder trata de suministrar energía contínua a la PC. La potencia iguala al voltaje por la corriente. Si el voltaje baja y se debe mantener la potencia constante, utilizará más corriente. Utilizar más corriente a través de un ccnductor hace que se caliente. La fuente de poder y los chips se calientan pudiéndose sobrecalentar. Los protectores contra alzas no ayudan en este caso. Los acondicionadores de potencia sí, ya que utilizan transformadores para compensar el voltaje fluctuante.

Descarga electrostática.- La descarga electrostática, más conocida como electricidad estática, se genera y acumula con mayor facilidad cuando el aire está muy seco, ya ue baja la humedad relativa y el aire es muy aislante. La electriciad estática se va generando y se acumula en la persona. Cuando la humedad relativa es alta, tambien se

genera y acumullan cargas estáticas, pero en este caso se disipan rápidamente debido a la humedad del aire. La resistencia de la piel tien mucho que ver con la disipación de las cargas. Esta resistencia puede ser tan baja como 1.00

página 2

ohms., estando húmeda, y hasta 500.00 ohms., estando seca. La estática puede dañar los chips si crea cargas de 1.000 volts ó mas. Si una descarga estática es lo suficientemente elevada para que la persona la note, es de 3.000 volts. Afortunadamentte el amperaje y la potencia son muy pequeños (La potencia es igual al voltaje por la corriente). Los materiales tienen un valor triboeléctrico. Los diversos materiales generan ,más ó menos estática. Dos materiales que se frotan entre sí generan estática en proporción directa a la diferencia en el valor triboeléctrico de cada uno. Se debe descargar la electriciad estática en algo metálico que no sea la cubierta de la computadora. Un escritorio metálico ó pata de mesa es bueno. Los chips que constituyen la mayor parte de las tarjetas de circuíto son dispositivos que se pueden dañar por el alto voltaje, aún a bajas corrientes. Evite trabajar sobre alfomabras (a menos que sean antiestáticas). No use sweaters acrílicos al cambiar chips. Utilice zapatos con suela de cuero. Si es posible, se puede evitar la estática quitándose los zaapatos y los calcetines. Toque los componentes sólamente por su parte no metálica. No toque las patas más de lo necesario. En todo caso, rocíe las alfombras con un rociador formado por un poco de suavizador para ropa y agua, bien agitado. Tambien se puede utilizar una tira metálica que consiste en una pulsera eléctrica, con una placa metálica interconstruída que suministra buena ccnexión eléctrica conectada a un conductor mediante un clip de caiman. Se pone el clip en algo que está a tierra y se sujeta la pulsera alrededor de la muñeca.

Corrosión.- Los líquidos y los gases pueden acelerar la corrosión de las PC y sus compoenntes. Entre estos agentes nocivos tenemos "La sal del sudor". Cuando los conectores de un aparato se oxidan, no conducen tan bien, y el parato no funciona ó funciona mal esporádicamente. La sal del sudor puede producir ésto, así que debe tenerse cuidado al manejar las tarjetas de circuíto. No toque los bordes a menos que sea indispensable. Los conectores de oro no se oxidan. Existen resíduos aceitosos en sus dedos. Hage una prueba. Vierta cerveza en un vaso muy limpio. Habrá bastante espuma sobre la bebida. Ahora meta un dedo en el centro de la espuma, sólo durante un segundo. La espuma se disuelve rápidamente, debido a que los aceites dañan la tensión superficial requerida para mantener la espuma. Los líquidos carbonatados tiene ácido carbónico, el café y el té contienen ácidos tánicos. El azúcar de los refrescos es atacada por bacterias que dejan resíduos de excremento conductor. Mojar una tarjeta sería como contratar gérmenes para que le coloquen nuevos conductores en las tarjetas de circuíto. Como regla general, evite los líquidos cerca de la computadora.

Magnetismo.- Los imanes, tanto los permanentes como los de tipo electromagnéticos, pueden causar la pérdida de los datos en los discos duros y diskettes. La mayoría de las fuentes de magnetismo son sobradamente conocidas. Algunas fuentes, menos conocidas y no menos importantes ,lo forman los monitores de rayos catódicos, y los teléfonos que tienen

página 3

campana mecánica (en vez de beep electrónico) , ya que el martillo es movido contra la campana por medio de un electroimán. Por lo tanto consiga un teléfono de llamador electrónico, es decir que emita beeps. Otra fuente importante de magnetismo es el motor de la impresora, que por lo general no está protegido. Los motores de las unidades de disco no producen mucho magnetismo. Asimismo hay que tener cuuidado con las máquinas de rayos X y los detectores de metales de los aeropuertos, ya que tambien producen magnetismo.

Interferencias electromagnéticas.- Se produce cuando el electromagentismo es radiado ó conducido a sitios donde puede causar interferencia. Los dos tipos más frecuentes son las líneas cruzadas y la interferencia de radio frecuencia. Las lineas cruzadas se producen cuando dos conductores están físicamente cercanos uno dee otro, ya que pueden producir interferencias entre éllos. Esto no significa cortocircuítos, el aislamiento puede estar en buen estado, El problema es que el conductor que causa la interferencia contiene impulsos electronicos, los cuales producen campos magnéticos como efecto colateral. Los pulsos electrónicos que se crean en el segundo cable son copias ténues de los pulsos, esto es , de la señal del primer conductor. Utilice pares trenzados. Variar el número de torceduras reduce el efecto de cables cruzados. Utilice cables blindados, ya que el blindaje reduce el cruce de cables. Utilice cables de fibra óptica, el cual es fotónico, no electromagnético, y por lo tanto en él no existe posibilidad de cruzamiento. No pase los cables sobre luces fluorescentes. No utilice cables de tipo listón para distancias mayores de 6 piés. Las interferencias de radio frecuencia son radiaciones de alta frecuencia (de 10 Khz y más), y constituye un problema serio. La RFI es mala porque puede interferir con los circuítos de alta velocidad, y la computadora está compuesta de circuitos digitales. Suponga que existe interferencia de radio flotando en el aire. No hay peligro mientras no se la capte. Pero, si por ejemplo, la computadora está conectada a una impresora con un cable que, por mala suerte, resulta ser la longitud correcta para captarla, tendremos problemas en la impresora. Afortunadamente, la solución es fácil: Acortar el cable. Tengamos en cuenta que la señal de radio frecuencia se recibe con una antena. Básicamente , la mejor longitud de esta antena, es una cuarta parte de la longitud de onda.

El rayo.- Cuando el martillo de Thor cae cerca de usted, no necesita equipo especial para notarlo. Las tarjetas de circuíto negras y torcidas son fáciles de notar. Estos son algunos puntos para entender cómo le afecta el rayo. El rayo afecta su sistema aún si no cae en su edificio. Es bueno dejar enchufada la máquina, ya el rayo tiene una ruta fácil a tierra. Tener especial cuidado de conectar a tierra la parte de edificio en dónde están las computadoras grandes sólo las hace más vulnerables. Una ruta con buena tierra es la que el rayo tomará. Los apartarrayos pueden reducir la posibilidad de daño por rayo. En algunos sitios se están usando

página 4

pararrayos nuevos de alta tecnología. Son como una sombrilla construída de alambre de púas como a 30 piés de altura. Un método barato de protección contra rayos se obtienee haciendo nudos en el cable eléctrico. Esto hace que la carga del rayo trabaje contra sí mismo y queme el cable, no la PC. Y por último, los aislantes ópticos protegen del rayo. Cortan la conexión eléctrica y transfieren los datos por medios ópticos

Nota: Toda esta información ha sido recogida y extractada de un interesante libro de Mark Minasi.

Félix Alonso
Burgos, Junio 1995

== APUNTES VARIOS DE INFORMATICA ==

FUENTE DE ALIMENTACION.- La entrana es de 220 v. Tiene una salida a +5 v (cable rojo) para las conexiones digitales en las placas del ordenador y otra de +12 V (cable amarillo) para los motores de los disquettes y discos duros. Los AT se

proveen de fuentes de alimentación de 200 y 220 W.

TARJETAS DE EXPANSION.- Se conectan a través del zócalo con el bus de datos procedente de la placa madre. El bus de datos se refiere a los conductos de la placa que transmite las informaciones hacia las diferentes partes del ordenador. También la CPU está conectada a éste bus de datos, de manera que el procesador de tarjetas pueda captar la información que necesita. Además las tarjetas están alimentadas a través de los zócalos con energía procedente de la fuente de alimentación.

La dirección de arranque, en el caso de tarjetas flexibles (tarjetas de expansión RAM), se ajusta normalmente mediante interruptores DIP que se encuentran situados sobre la placa de expansión, es decir a través de éllos se ajsta cuanta memoria RAM se encuentra instalada sobre la placa madre y cuanta en la tarjeta de expansión.

BUSES ESTANDARD ISA, EISA Y MCA.- El bus MCA no es compatible con el standard ISA, es decir que las tarjetas de expansión ISA no pueden ser instaladas en los PS/2 ó viceversa. La compatibilidad de las tarjetas de expansión según ISA standard está plenamente garantizada con los buses EISA. Mediante la construcción de las conexiones de dos niveles, pueden ser instaladas tanto tarjetas ISA como también EISA indistintamente en la misma conexión.

La frecuencia de reloj del bus EISA está sincronizada con la del sistema, lo cual permite, con respecto a la arquitectura MCA, una transmisión de datos más elevada (en un factor de 1.3).

Mientras en el bus ISA se adjudica a cada tarjeta de expansión una señal de interrupción fija, las tarjetas EISA pueden compartir una misma señal de interrupción mediente una técnica especial.

Al igual que con el sistema MCA también en el bus EISA el espacio de dirección reservado para una tarjeta depende de la conexión en la que ésta esté instalada.

FRECUENCIA DEL CUARZO Y FRECUENCIA DEL SISTEMA.- La CPU trabaja con la mitad de la frecuencia del cuarzo, de manera que un cuarzo de 24 Mhz produce una frecuencia de reloj del sistema de 12 Mhz. Un AT de 12 Mhz, consecuentemente, está equipado por regla general con 3 cuarzos: Uno para la frecuencia del bus de 4,77 Mhz (frecuencia de cuarzo correspondiente de 14,31818 Mhz); Uno para la frecuencia de trabajo normal de 6 Mhz (frecuencia de cuarzo correspondiente de 12 Mhz): y finalmente uno para la frecuencia de trabajo "Turbo" ó "Hightspeed" de 12 Mhz (frecuencia de cuarzo correspondiente de 24 Mhz).

A través de la sustitución del cuarzo de 24 Mhz por otro de 32 Mhz., se podría aumentar la frecuencia del sistema en

página 1

"nivel turbo" a 16 Mhz.

En el XT el generador de frecuencia es responsable tanto de la CPU como del bus. En el AT ésta tarea se lleva a cabo con diferentes cuarzos. La frecuencia del bus de una placa AT es producida, al igual que en un XT, por un cuarzo con una oscilación característica propia de 14,31818 Mhz.

El cuarzo de 14,31818 Mhz se encuentra dividido en 3 canales timer de 4,77 Mhz (14,31818:3=4,77) cada uno, lo cual resulta exactamente la frecuencia normal de reloj de un XT. Al mismo tiempo, también trabaja con la misma frecuencia el bus en el cual están conectadas las tarjetas de ampliación. Los adaptadores de vídeo, por ejemplo, producen con élla su frecuencia de modulación de color.

CONFIG.SYS.- BUFFERS. Con la línea BUFFERS=15 instalará un BUFFERS de 7,5 Kby en la memoria principal para los accesos al disquette ó bien al disco duro. FILES. El valor máximo es 255 y valores inferiores a 8 no son tenidos en cuenta. Cada fichero que se añada a los 8 (valor inicial a que está ajustado), emplea una memoria adicional de 48 Kby.

Félix Alonso

Burgos, Noviembre 94

========= NOMBRE DE DISPOSITIVOS =========

En este pequeño artículo voy a tratar de completar, con una exposición más detallada y comentada, la información sobre este tema contenida en el libro de A. DICKENS.

A.- Descripción del nombre del Dispositivo.

La descripción completa del nombre del Dispositivo (ROM del SISTEMA), presenta el siguiente formato:

```
     1.- Número de caracteres en el nombre (palabra),
         caracteres ASCIL del nombre (palabra).
         Ejemplo:      DC.W  3, "CON"
     2.- Número de parámetros (palabra).
         Ejemplo:      DC.W  5
     3.- Por cada parámetro, una de las trés formas siguien-
         tes:
         a).- Espacio + separador, valor por defecto (numéri-
              co). Es decir si no se da el  valor,
              el Sistema devuelve un valor por defecto.
              Ejemplo:  DC.W  "_", 448
                        DC.W  "x", 200
         b).- Número negativo, valor por defecto (número,
              no separador). Igualmente si no se da el va-
              lor, el Sistema devuelve un valor por defec-
              to.
              Ejemplo:  DC.W  -1, 1
         c).- Número de código, lista de códigos ASCIL. Si no
              se indica ninguna, el Sistema devuelve 0.
              Ejemplo:  DC.W  4, "EOMS". Posición 4 en la lis
              ta "EOMS".
```

Cómo vemos todos los ítems se definen como palabra. Todas las letras deben ir en mayúsculas. Comoquiera que todos los códigos ASCIL de los caracteres ocupan sólamente un octeto, cualquier octeto impar al final, debe rellenarse con cero.

Ejemplo: DC.W "ABC". Aparecerá en memoria de la siguiente forma:

```
                  $00 03  Palabra primera
                  $41 42  Palabra segunda
                  $43 00  Palabra tercera
```

La descripción completa de CON (ROM del SISTEMA) es la siguiente:

```
     DC.W  3, "CON"            Consola, 3 caracteres
     DC.W  5                   5 parámetros
     DC.W  "_",448, "x", 200   Tamaño ventana. 2 parámetros.
                               Separadores y valores numéri-
                               cos.
     DC.W  "a", 32, "x", 16    Posición ventana. 2 Parámetros
                               Separadores y  valores numéri-
                               cos.
     DC.W  "_", 128            Longitud  memoria  intermedia
                               del teclado. 1 parámetro. Se-
                               parador y valor numérico.
```

página 1

La descripción completa de SER (ROM del SISTEMA) es la siguiente:

```
     DC.W  3, "SER"           Dispositivo serie RS232
     DC.W  3                  3 parámetros
     DC.W  -1, 1              Número de puerta. 1 por defec-
                              to. Número negativo.
     DC.W  4, EOMS"           Paridad impar, par, marca ó es-
```

```
                         pacio. Código de lista.
DC.W  2, "IH"            Ignorar ó usar "handshaking".
                         Código de lista.
```

La descripción completa de SCR (ROM del SISTEMA) es la siguiente:

```
DC.W  3, "SCR"           Dispositivo pantalla
DC.W  4                  4 parámetrso
DC.W  "_", 448, "x", 180 Tamaño ventana. 2 parámetros.
                         Separadores y valor numérico.
DC.W  "a", 32, "x", 16   Posición ventana. 2 parámetros.
                         Separadores y valor numérico.
```

B.- Decodificación del nombre del Dispositivo.

El nombre completo del Dispositivo contiene cuatro elementos:

1.- Nombre. Caracteres ASCIL, normalmente letras independientemente de si son mayúsculass ó minúsculas.

2.- Separador. Caracteres ASCIL. Si es una letra es independiente de si es mayúscula ó minúscula.

3.- Número. Número decimal en el rango de 0 a 2^15-1.

4.- Código.- Uno de la lista de caracteres ASCIL.

La decodificación del nombre del Dispositivo facilitado por el usuario, se lleva a cabo con la rutina IO.NAME (vector $122), la cual realiza dos operaciones:

1.- Comprobar el nombre del Dispositivo.

2.- Evaluar cualquier parámetro opcional que encuentre.

Por cada parámetro numérico de la descripción del Dispositivo, la rutina devolverá el valor dado por el usuario, ó el asignado por defecto, si no se ha facilitado ninguno.

Por cada lista de códigos de la descripción del Dispositivo, la rutina devolverá la posición indicada en la lista, ó cero, si no se facilita este dato.

C.- Utilización de la rutina IO.NAME.

Cuando se llama a la rutins IO.NAME (vector $122), los siguientes registros deben contener estos datos:

```
  A0 = Apuntador al nombre del Dispositivo.
  A3 = Apuntador a un bloque de memoria intermedia que
       sea suficiente para almacenar los valores de los
       parámetros.
```

Esta rutina tiene trés possbles retornos:

```
       Retorno      DO(+SR)
  a).- Estandar     ERR.NF.   Nombre no reconocido.
  b).- Estandar+2   ERR.BN    Nombre reconocido, pará-
                              metro malo.
                        página 2
  c).- Estandar+4   0. Retorno satisfactorio.
```

La descripción del nombre del Dispositivo comienza séis octetos después de la llamada, es decir después de los trés posibles retornos.

D.- Veamos a continuación algunos ejemplos de descripción de nombres de Dispositivos, debidamente comentados:

```
Nombre  Dispositivo  Parámetros devueltos por el sistema

CON                  448,200,32,16,128. Todos asignados
                     por defecto, por omisión de pará-
                     metro numérico.

CON_256              256,200,32,16,128. El primer pará-
                     metro después del primer separador
                     "_", ha sido facilitado (256). Los
                     restantes son asignados por defec-
                     to.

con_60               448,200,32,16,60. Los  cuatro pri-
```

meros parámetros han sido asignados por defecto. El último parámetro, después del segundo separador "_", ha sido facilitado (60).

cona0x12 — 448,200,0,12,128. Los dos primeros parámetros han sido asignados por defecto. Los dos siguientes, después del separador "a", han sido facilitados. El último ha sido asignado por defecto.

SER — 1, 0, 0. Todos los parámetros asignados por defecto. El primero por omisión de parámetro numérico. Los otros dos por omisión de código.

sere — 1, 1, 0. El primer parámetro asignado por defecto, por omisión de parámetro numérico. El segundo ha sido facilitado y corresponde al código en la posición 2 de la lista "EOMS". El tercero ha sido asignado por defecto, por omisión de código.

ser2mi — 2, 3, 1. El primer parámetro ha sido facilitado, valor numérico. El segundo ha sido también facilitado y corresponde al código en la posición 3 de la lista "EOMS". El tercero asimismo ha sido facilita-
página 3
do y corresponde al código en la posición 1 de la lista "IH".

Félix Alonso
Burgos, Diciembre 1995

GLOSARIO DE TERMINOS INFORMATICOS

ABOVE-BOARD.- Tarjeta de expasión de memoria que amplía la memoria de trabajo de la placa madre. Se puede configurar mediante un programa de instalación en memeoria "Base", "Expandida" y "Extendida".

ARRANQUE EN CALIENTE.- Vuelta al principio y nuevo arranque del sistema mediante las teclas CTRL+ALT+DEL.

AT.- Advanced Technology.

BTT.- Bad Track Table. Indica los sectores marcados como defectuosos en el formato low_level.

BANK SWITCHING.- Conmutación de un sector de memoria por otro.

BEEP CODE.- Señal acústica que emite el PC durante la rutina de chequeo automático Power_on_self_test (POST) a través de un altavoz.

CARTRIDGE.- Soporte de memoria removible que se utiliza como soporte de datos de las unidades de cinta (Streamers).

CMOS.- Chips especialmente desarrollados para ahorrar energía eléctrica, sobre todo en laptops (ordenadores portátiles). Funcionan mediante la utilización de baterías de mentenimiento. Permiten guardar la información que contienen con una alimentación mínima. Se utilizan para la grabación de la configuración del sistema de ordenadores del tipo AT.

CONEXION KEYLOCK.- A través de él y controlado mediante una cerradura es posible bloquear el teclado. Es un circuito

de conexiones que se encuentra en muchas de las placas mader modernas.

CPI.- Characters per inch. Caracteres por pulgada.

CPS.- Characters per second. Caracteres por segundo.

DIP.- Dual Inline Package.

DMA.- Direct memory acces. Acceso directo a la memoria.

DOP PITCH.- Distancia entre los diferentes puntos de imagen en la pantalla de un monitor.

DPI.- Dots per inch. Puntos por pulgada. Unidad de medida de la resolución gráfica de aparatos periféricos, como impresora, scanner, etc.

DTP.- Desk Top Publishig. Pupitre de autoedicción para PC.

EISA.- Extended Industry System.

ESDI.- Enhanced Small Device Interface. Los discos duros ESDI emiten y reciben datos digitales. La controladora no precisa efectuar una transformación analógica_digital. La transmisión de datos es más rápida y segura en un factor 2,5 que con el standard usual ST-506.

FAT.- File Allocatión Table.

FILE_SERVER.- Distribuidor de ficheros en la RED.

FORMATO LOW LEVEL.- Inicialización ó formato preliminar.

HARRIS.- Juego de chips para las placas madre de sistemas 80286 con Tecnología NEAT.

HEAD-CRASH.- Destrucción de la superficie del disco duro debido a la precipitación del cabezal de lectura/escritura.

IDE.- Dispositivo electrónico integrado. Controlador de disco duro instalado en su misma unidad. Necesita sólo una

página 1

tarjeta sencilla de adaptación al Bus.

INBOARD.- Placa de ampliación que completa a la placa madre con una CPU más potente.

INTERRUTOR DIP (DIP SWITCH).- Dual Inline Package. Micro interruptor que se encuentra en placas, tarjetas, ampliaciones, etc., para definir posiciones básicas.

KEYBOARD.- Teclado.

LAPTOP.- Ordenador portátil.

LCD.- Liquid Crystal Display. Pantalla de cristal líquido.

MODERBOARD.- Placa madre.

NLQ.- Near Letter Quality.

OFFLINE.- Estado no disponible para la recepción de datos.

ONLINE.- Estado disponible para la recepción de datos.

PAGE FRAME.- Ventana de memoria de la EMS.

PARAMETROS DE DISCO DURO.- Datos técnicos para la instalación del disco duro: número de cabezas, pistas, sectores, etc.

PARTICION.- Unidad lógica del disco duro.

PORT.- Puerto de conexión.

POST.- Power On Self Test. Rutinas de diagnosis que se realizan automáticamente.

POWER SUPPLY.- Fuente de alimentación del PC.

MFM.- Modified Frecuency Modulación. Procedimiento de grabación muy fiable pero que emplea mucho espacio de memoria.

RAM.- Random Acces Memory.

REFRESH.- Renovación ó refresco del contenido de la memoria de chips RAM dinámicas (DRAMS) antes de que se pierdan ó deterioren los datos.

RGB.- Red Green Blue. Los tres colores básicos: Rojo, verde y azul.

RLL.- Run Lengh Limited. Procedimiento de grabación.

ROM.- Read Only Memory.

ROOT.- Directorio principal en el disco duro, que, aparte del Command.com contiene tambien los ficheros ocultos del Sistema Operativo.

SCAN CODE.- Señal codificada de manera electrónica que es transmitida por el procesador de teclado a la CPU para que ésta reconozca que tecla ha sido pulsada.

SCSI.- Small Computer System Interface. Standard de conexiones para discos duros y controladores, aí como de otros dispositivos periféricos. Los disocs duros del tipo SCSI integran las funciones de la controooladora en la misma electrónica del disco duro.

SETUP.- Programa para el ajuste de la configuración del Hardware de un AT ó de un 386. Estos ajustes se graban en una memoria RAM del tipo CMOS alimentada con una batería.

SIP.- Single Inline Package.

SIMM.- Single Inline Memory.

SPOOLER.- Dispositivo para la grabación temporal de datos (BUFFER).

SPEEDSTAR.- Programa para la inicialización de discos

duros de la Empresa STORAGE DIMENSIONS.

TECNOLOGIA NEAT.- New Enhanced Advanced Technology.

TRACK.- Pista.

TTL.- Transistor Transistor Logic. Tipo de monitores.

WAIT STATES.- Ciclos de espera.

XCMOS.- Extended CMOS. CMOS en tecnología NEAT y en ordendores del tipo 386.

XT.- Extended Technology.

Félix Alonso
Burgos, Noviembre 94